

STIMULATING SIMULATIONS

Ten unique programs in BASIC
for the computer hobbyist

by

C. William Engel

Dr. Engel is a professor of Mathematics
Education at the University of South Florida
Tampa, Florida

Editor: Jane E. Engel

Consultants: Sandra J. Engel
Gregory C. Engel

Published by
C. William Engel
Box 16612
Tampa, Florida 33687

Copyright © C. William Engel 1977

All rights reserved.

No part of this book may be
reproduced by any means without the
written permission of the publisher.
Programs may be stored and retrieved
electronically for personal use.

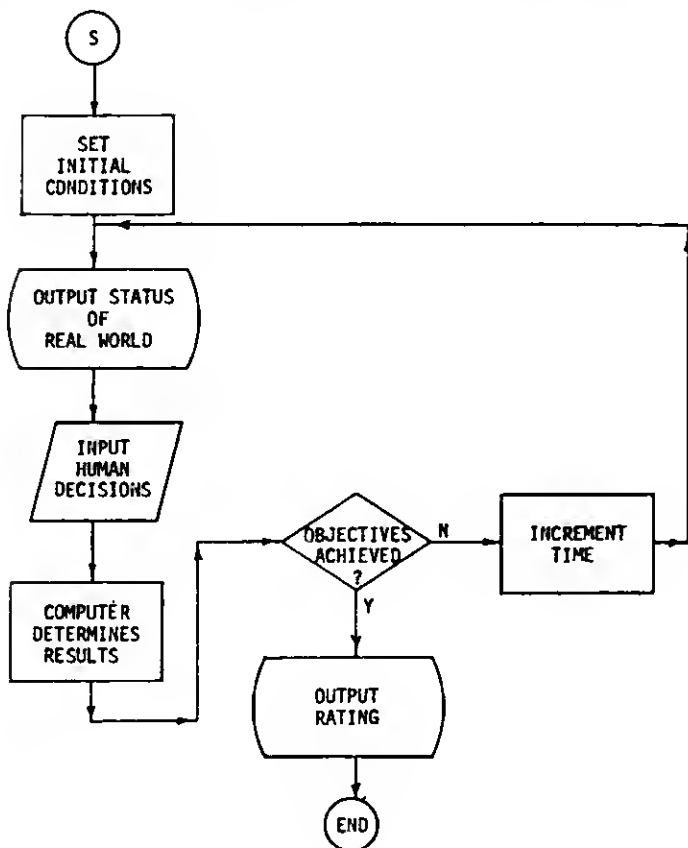
CONTENTS

INTRODUCTION	1
ART AUCTION	1.1
MONSTER CHASE	2.1
LOST TREASURE	3.1
GONE FISHING	4.1
SPACE FLIGHT	5.1
FOREST FIRE	6.1
NAUTICAL NAVIGATION	7.1
BUSINESS MANAGEMENT	8.1
RARE BIRDS	9.1
DIAMOND THIEF	10.1

INTRODUCTION

Simple number games and puzzles are used frequently by beginning computer hobbyists. While some computer enthusiasts develop computer systems that monitor environmental conditions, compute income tax, or serve as expensive burglar alarms, most continue to use their computers primarily for recreation. This booklet is designed for the person who is beyond the simple number-game stage of software development and would like to develop some interesting simulations.

The programs are written so that the computer does not do all the "thinking" but forces the player to develop strategies for achieving the objectives. A general overview of a simulation is illustrated in the flowchart below.



The simulations presented in this booklet are written in BASIC and can be easily adapted to almost any system. The programs vary from 500 to 2,000 bytes or 40 to 100 lines of BASIC. Some of the lines have multiple statements; but, since the line numbers are multiples of ten, it would be easy to modify the program to operate with single statements. All of the line numbers with a unit's digit of five can be deleted without affecting the program.

Each simulation begins with a scenario describing the rules, conditions and objectives to be achieved. The rules have been written in third person, because some programmers like to condense the rules and place them in a subroutine for access by the operator. A sample run and a general flowchart with line numbers provide additional information about each program. A description of the variables precedes the program listing. Some program modifications are suggested. The minor modifications require only adjustments of variables in specific lines, while major modifications require additional programming. In some cases, supplemental playing boards, graphs, and charts are supplied for recording information on the progress of the simulation.

A brief description of each program is given below.

1. Art Auction (48 lines)
One buys and sells paintings to make a maximum profit. This is a fast simulation and does not require extra materials.
2. Monster Chase (48 lines)
A monster is chasing a victim in a cage. The victim must elude the monster for ten moves to survive. It is a fairly quick simulation that doesn't require too much thinking.
3. Lost Treasure (74 lines)
A map of an island that contains treasure is presented. The adventurer travels over different terrain with a compass that isn't very accurate in an attempt to find the treasure. This is a short simulation that requires about 15 moves. A map is provided.
4. Gone Fishing (83 lines)
The objective is to catch a lot of fish during a fishing trip. Half of the catch spoils if the time limit is exceeded, time is lost in a storm, and the boat sinks if it is guided off of the map. There are also sea gulls and sharks to watch. A chart is needed to keep track of good fishing spots.
5. Space Flight (68 lines)
The task is to deliver medical supplies to a distant planet while trying to stay on course without running out of fuel. Graph paper is required to plot the course.
6. Forest Fire (77 lines)
The objective is to subdue a forest fire with chemicals and backfires. Because the output is a

9X9 grid, a fast baud rate to the terminal is desirable. The success of a firefighter is based on the time needed to control the fire and completely extinguish it.

7. Nautical Navigation (70 lines)
This simulation requires the navigation of a sailboat to three different islands, using a radio direction finder. The wind direction is an important variable. Graph paper, protractor and ruler are needed to plot the course.
8. Business Management (92 lines)
In this simulation, raw materials are bought and finished products are produced and sold. The cost of materials and production and the selling price vary each month. The objective is to maximize the profits. No extra materials are required.
9. Rare Birds (75 lines)
This is a bird watching simulation. The objective is to identify as many different birds as possible. A record of those identified is helpful and a bird watching chart is provided.
10. Diamond Thief (83 lines)
One assumes the role of a detective in this simulation. A thief has just stolen a diamond from a museum. Five suspects must be questioned to determine the thief. A floor plan of the museum and a chart indicating suspects and times are provided.

In addition to extending the simulations in this booklet, one might try combining some of them. For example: one could take the money earned in Art Auction to start the Business Management simulation. After twelve months of business, the profits could be used to buy a boat to use in the Gone Fishing simulation. A large boat could survive storms, hold more fish, and allow fishing in deeper water. The ultimate objective could be to catch the most fish.

The computer hobbyist is limited only by the imagination in simulating real events. It is the author's desire that this booklet provide some fun and, at the same time, stimulate further development of creative simulations. Some additional ideas for simulations are suggested below:

1. Hunt Big Foot
2. Race a Sailboat
3. Inhibit the Andromeda Strain
4. Stop the African Bee Invasion
5. Climb Mountains
6. Survive in the Wilderness
7. Find Gold or Oil
8. Swim from Jaws
9. Dispatch Airplanes, Trains, or Trucks
10. Herd Sheep
11. Explore Caves
12. Catch Butterflies

ART AUCTION

Scenario

In this simulation, you will be given an opportunity to buy and sell up to five paintings. The objective is to make a large profit by buying the paintings for as little as possible and selling them for as much as possible.

In order to buy a painting, you must bid against a secret bid made by another buyer. When a painting is offered for sale, three numbers will be given that represent the mean and range of bids for this particular painting. For example, "200 300 400" indicates that the mean bid price for the painting is 300, and about 70% of the time the price will be between 200 and 400. (Note that higher priced paintings tend to have a larger range of prices.)

After you buy your paintings, you will be given an opportunity to sell them. You will receive from one to five offers, but you do not know in advance how many offers will be made. The offers will be, on the average, 50 higher than the bids made during the buying phase. If you do not accept an offer, and it is the last one, then the offer will be automatically processed. Sometimes it will be wise to accept an offer that is less than the purchase price rather than gamble on a higher offer that does not materialize.

When all of the paintings that you have bought have been sold, you will be given your total profit for all of the transactions.

Sample Run

BUY PAINTING 1
PRICES: 546 553 560
YOUR BID? 560
OPPONENT BID 565.
YOU WERE OUT B10.

BUY PAINTING 2
PRICES: 336 449 562
YOUR BID? 400
OPPONENT BID 440.
YOU WERE OUT B10.

BUY PAINTING 3
PRICES: 213 288 363
YOUR BID? 300
OPPONENT BID 324
YOU WERE OUT B10.

BUY PAINTING 4
PRICES: 403 514 625
YOUR BID? 600
OPPONENT BID 497.
YOU BOUGHT IT.

BUY PAINTING 5
PRICES: 274 346 417
YOUR BID? 350
OPPONENT BID 311.
YOU BOUGHT IT.

SELL PAINTING 4
YOU BOUGHT IT FOR 600.
AVERAGE OFFER IS 564.
OFFER 1 IS 649.
ACCEPT? Y

SELL PAINTING 5
YOU BOUGHT IT FOR 350.
AVERAGE OFFER IS 396.
OFFER 1 IS 365.
ACCEPT? N

YOUR PROFIT IS 64.
PLAY AGAIN?

ART AUCTION PROGRAM

Variables

P(5)	Prices
S(5)	Price range
F(5)	Set flag if painting is bought
CB	Opponent's bid
YB	Your bid
I,J,K	Indices
P	Profit
N	Number
O	Dividend
Q	Quotient

Program Listing

```

5      REM SET PRICES AND RANGES
10     DIM P(5),S(5),F(5)
20     FOR I=1 TO 5
30     P(I)=100+INT(900*RND(1))
40     S(I)=INT(P(I)*RND(1))
50     IF P(I)<500 THEN S(I)=INT(P(I)*.7*RND(1))
60     F(I)=0
70     NEXT I

95     REM BUY PAINTINGS
100    FOR I=1 TO 5
110    GO SUB 500
120    PRINT: PRINT "BUY PAINTING"; I:PRINT:PRINT
130    PRINT "PRICES:"; INT(P(I)-.5*S(I)); P(I); INT(P(I)+.5*S(I))
140    PRINT: PRINT: INPUT "YOUR BID"; YB
150    PRINT "OPPONENT'S BID"; CB; "."
160    IF YB>CB THEN PRINT "YOU BOUGHT IT."; F(I)=YB: GO TO 180
170    PRINT "YOU WERE OUT BID."
180    NEXT I

195    REM SELL PAINTINGS
200    FOR I=1 TO 5
210    IF F(I)=0 THEN 310
220    FOR K=1 TO INT(5*RND(1))
230    GO SUB 500: CB=CB+INT(100*RND(1))

240    PRINT "SELL PAINTINGS"; I
250    PRINT "YOU BOUGHT IT FOR"; F(I): PRINT "AVERAGE OFFER IS";
      P(I)+50
260    PRINT "OFFER"; K; "IS"; CB; "."
270    INPUT "ACCEPT"; Y$
280    IF Y$="Y" THEN 300
290    NEXT K
300    P=P+CB-F(I)
310    NEXT I
320    PRINT: PRINT "YOUR PROFIT IS"; P; "."
330    INPUT "PLAY AGAIN"; Y$
340    IF Y$="Y" THEN RUN
350    END

```

```

495 REM NORMAL DISTRIBUTION SUBROUTINE
500 D=0
510 N=INT(65536*RNO(1))
520 FOR J=1 TO 16
530 Q=INT(N/2)
540 D=D+2*(N/2-Q)
550 N=Q
560 NEXT J
570 CB=P(I)+5(I)*(D-B)/B
580 CB=CB+20*RNO(1)
590 CB=INT(CB)
600 RETURN

```

ART AUCTION MODIFICATIONS

Minor

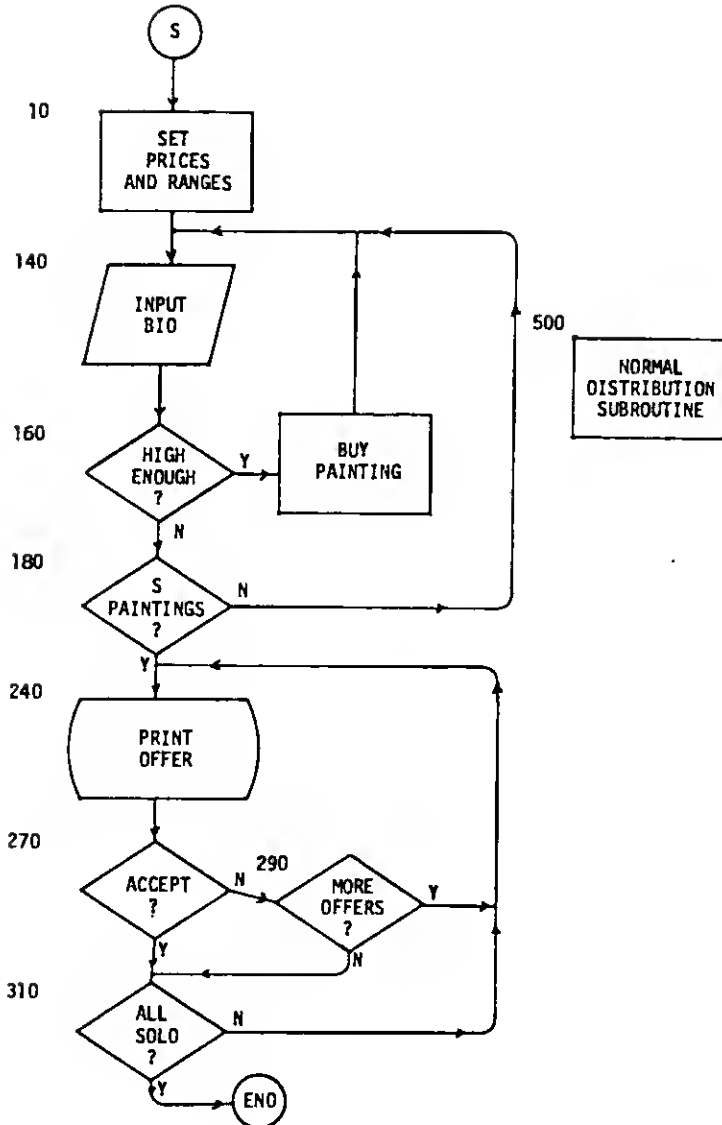
1. Number of paintings -- lines 10, 20, 100, 200
2. Starting prices -- line 30
3. Price spread -- lines 40, 50
4. Built-in profit -- lines 230, 250
5. Error in price range -- line 580
6. Number of offers -- line 220

Major

1. Have one or more of the paintings a forgery that is worth nothing.
2. Have one or more of the paintings that have a low purchase price be very valuable.
3. Have more opponents bid against you.



ART AUCTION FLOWCHART

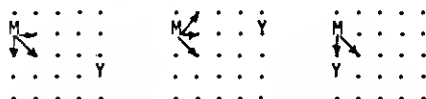


MONSTER CHASE

Scenario

In this simulation you are locked in a cage with a hungry monster who has a life span of ten turns. Your movement and that of the monster takes place on a 5X5 grid. You may move north, east, south, or west by entering N, E, S, or W. If you enter any other letter, you will remain in the same place.

The monster is programmed to move along one of the arrows toward you as shown below :



Your only means of survival is to outwit the monster for ten turns.

Sample Run

<pre> M . Y </pre>	<pre> M Y </pre>	<pre> Y M </pre>
MOVE 1 DIRECTION? W	MOVE 4 DIRECTION? W	MOVE 7 DIRECTION? W
<pre> . M Y </pre>	<pre> M Y </pre>	<pre> Y M </pre>
MOVE 2 DIRECTION? N	MOVE 5 DIRECTION? W	MOVE 8 DIRECTION? N
<pre> M Y </pre>	<pre> Y . Y . M </pre>	EATEN PLAY AGAIN?
MOVE 3 DIRECTION? S	MOVE 6 DIRECTION? N	

MONSTER CHASE PROGRAM

Variables

L(I,J)	Grid location
R,C	Your row and column
X,Y	Monster's row and column
L,M	Temporary variables
M\$	Your move (N,E,S,W,O)
D	Direction of the monster (1-8)
T	Turns (1-10)

Listing

```

5      REM SET CONOITIONS
10     X=1: Y=1
20     R=5: C=5
30     FOR T=1 TO 10

35     REM OI5PLAY GRIO
40     FOR I=1 TO 5
50     FOR J=1 TO 5
60     PRINT TAB(8)
70     IF I=X AND J=Y THEN PRINT "M";: GO TO 100
80     IF I=R AND J=C THEN PRINT "Y";: GO TO 100
90     PRINT ". ";
100    NEXT J
110    PRINT
120    NEXT I

210    ?;?:? "MOVE NUMBER": T
220    INPUT "DIRECTION (NESWO)": M$
240    IF M$="N" THEN R=R-1
250    IF M$="E" THEN C=C+1
260    IF M$="S" THEN R=R+1
270    IF M$="W" THEN C=C-1
280    IF R<0 OR R>5 OR C>5 THEN PRINT "OUT OF BOUNDS": GO TO 520
290    IF R=X AND Y=C THEN PRINT "EATEN": GO TO 520
300    IF X=R AND Y<C THEN D=1
310    IF X>R AND Y<C THEN D=2
320    IF X>R AND Y=C THEN D=3
330    IF X>R AND Y>C THEN D=4
340    IF X=R AND Y>C THEN D=5
350    IF X<R AND Y<C THEN D=6
360    IF X<R AND Y=C THEN D=7
370    IF X<R AND Y>C THEN D=8
380    D=D+INT(3*RND(1)-1)
390    IF D=0 THEN D=8
400    IF D=9 THEN D=1
410    IF D>1 AND D<5 THEN X=X-1
420    IF D>5 THEN X=X+1
430    IF D>3 AND D<7 THEN Y=Y-1
440    IF D<3 OR D=8 THEN Y=Y+1
450    IF X=0 THEN X=X+1
460    IF Y=0 THEN Y=Y+1
470    IF X=6 THEN X=X-1
480    IF Y=6 THEN Y=Y-1

```

```
490 IF X=R AND Y=C THEN PRINT "EATEN": GO TO 520
500 NEXT T
510 PRINT "YOU SURVIVED!"
520 INPUT "PLAY AGAIN?"; Y$
530 IF Y$="Y" THEN RUN
540 END
```

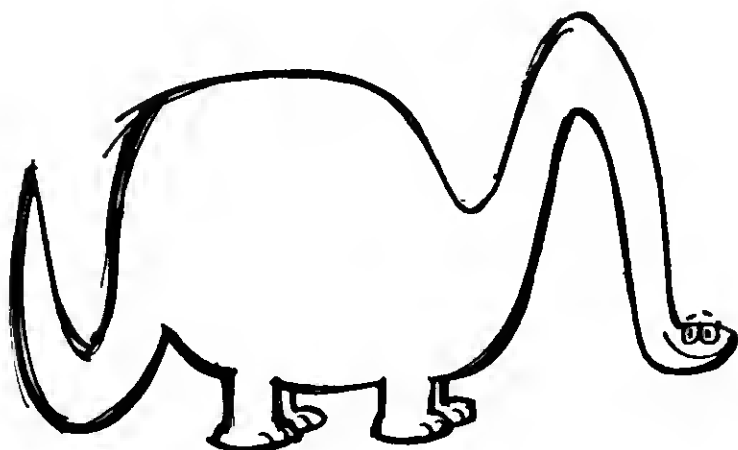
MONSTER CHASE MODIFICATIONS

Minor

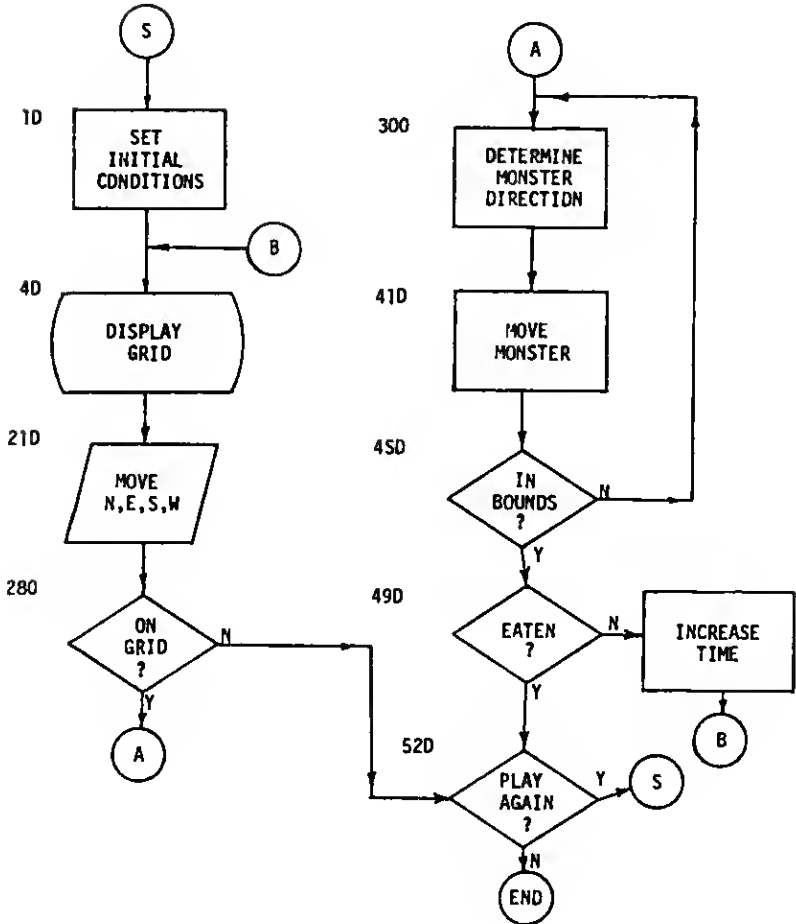
1. Grid size -- lines 20, 40, 50, 280, 470, 480
2. Turns to win -- line 30

Major

1. Have more than one monster.
2. Chase a little monster while a big monster tries to get you.
3. Have the monster fall in quicksand.
4. Require food in order to maintain energy.



MONSTER CHASE FLOWCHART



LOST TREASURE

Scenario

You have landed somewhere on an island that has treasure, woods, mountains, a cave, a bluff, an oak tree, and, of course, sea water all around. Your objective is to find the treasure as quickly as possible without falling into the shark-infested water.

You can move north (N), east (E), south (S), or west (W) one square at a time. Your compass, however, is not very accurate. There is only an 80% chance that you will move in the intended direction. There is a 20% chance you will move diagonally to the left or to the right. Each time that you move you will receive feedback regarding the type of terrain on which you are traveling.

If you fall into the sea, you will be placed back on the square occupied prior to your unfortunate move, unless you disturb the sharks. The chance that the sharks will eat you the first time you fall in is 20%. The second time you fall in the chance of being eaten is 70%. The third time you fall in will be your last!

Since you have a map of the island, you will be able to determine your approximate position. For example, if you are in the woods and you move east two squares and find that you are in mountains, then you are most likely located in the north-east corner of the island. The reason you can't be sure of the exact location is that you may have veered off to the right or left. With practice, you should be able to find the treasure in less than fifteen moves.

Sample Run

RUN

YOU ARE IN THE CLEAR.

MOVE(NESW)? S

YOU FELL INTO THE OCEAN.

EATEN BY SHARK.

PLAY AGAIN Y OR N? Y

YOU ARE IN THE CLEAR.

MOVE(NESW)? S

YOU ARE IN THE WOODS.

MOVE(NESW)? N

.
.
.

.
.

YOU ARE IN THE MOUNTAINS.

MOVE(NESW)? E

.
.
.

YOU ARE IN THE WOODS.

MOVE(NESW)? S

.
.
.

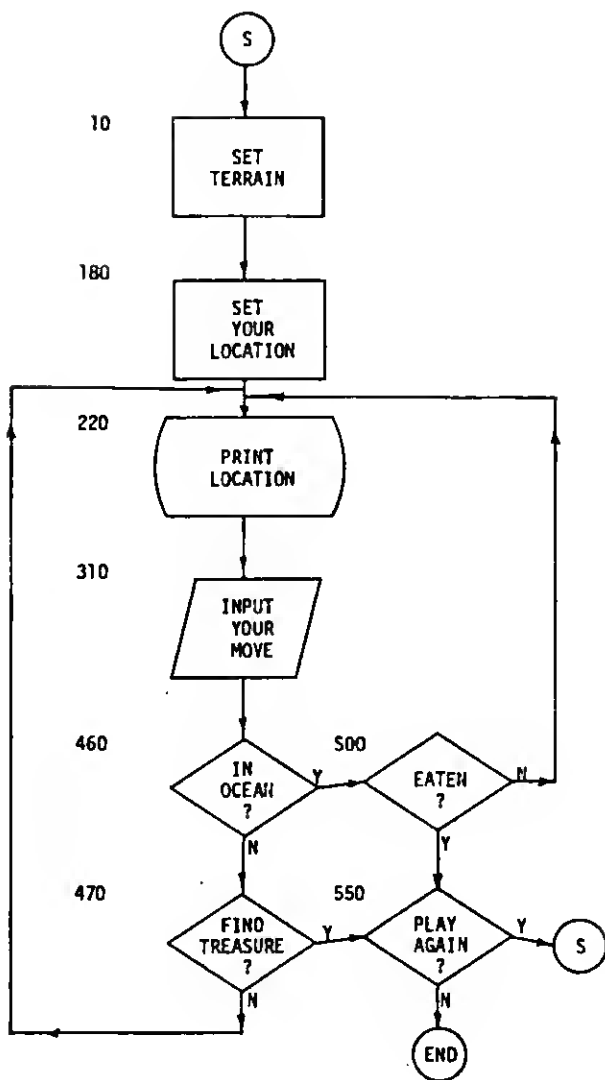
YOU ARE IN THE CLEAR.

MOVE(NESW)? E

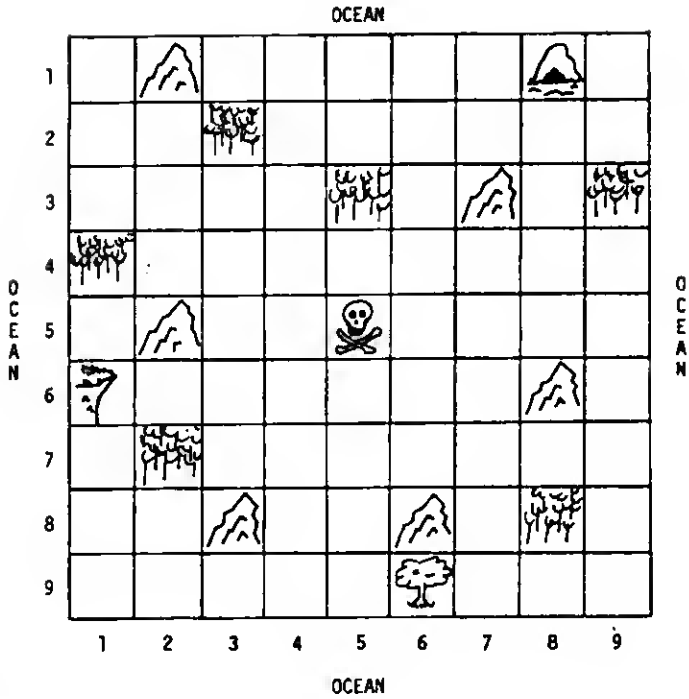
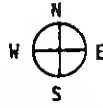
YOU FOUND THE TREASURE IN 9 MOVES.

PLAY AGAIN Y OR N?

LOST TREASURE FLOWCHART



LOST TREASURE MAP



Legend

Mountains

Oak Tree

Bluff

Woods

Cave

Treasure

LOST TREASURE PROGRAM

Variables

L(R,C) Locations
S Probability of being eaten by shark
R Your row
C Your column
RT, CT Temporary storage
T Number of turns

Listing

```

5   REM SET TERRAIN
10  DIM L(9,9)
20  S=.2
30  FOR I=1 TO 9: FOR J=1 TO 9
40  L(I,J)=0
50  NEXT J,I

60  FOR I=1 TO 6
70  READ R,C
80  L(R,C)=1
90  NEXT I

100 FOR I=1 TO 6
110 READ R,C
120 L(R,C)=2
130 NEXT I

140 L(1,8)=3
150 L(6,1)=4
160 L(9,6)=5
170 L(5,5)=6

175 REM YOUR LOCATION
180 R=INT(9*RND(1))+1
190 C=INT(9*RND(1))+1
200 IF SQR(R-5)+2+(C-5)+2)<2 THEN 180

205 REM START MAIN LOOP
210 FOR T=1 TO 100
220 PRINT "YOU ARE ";
230 J=L(R,C)+1
240 ON J GO SUB 250,260,270,280,290,300: GO TO 310
250 PRINT "IN THE CLEAR.": RETURN
260 PRINT "IN THE WOODS.": RETURN
270 PRINT "IN THE MOUNTAINS.": RETURN
280 PRINT "NEAR A CAVE.": RETURN
290 PRINT "ON A BLUFF.": RETURN
300 PRINT "NEAR AN OAK TREE.": RETURN

310 INPUT "MOVE(NESW)"; M$
320 RT=R: CT=C
330 IF M$="N" THEN R=R-1: GO SUB 380
340 IF M$="E" THEN C=C+1: GO SUB 420
350 IF M$="W" THEN C=C-1: GO SUB 420
360 IF M$="S" THEN R=R+1: GO SUB 380

```



```

370 GO TO 460

375 REM MOVE SUBROUTINE
380 J=INT(10*RND(1)+1)
390 IF J>2 THEN RETURN
400 IF J=1 THEN C=C+1: RETURN
410 C=C-1: RETURN
420 J=INT(10*RND(1)+1)
430 IF J>2 THEN RETURN
440 IF J=1 THEN R=R+1: RETURN
450 R=R-1: RETURN

455 REM IN OCEAN, FOUND TREASURE?
460 IF R<1 OR R>9 OR C<1 OR C>9 THEN 490
470 IF L(R,C)=6 THEN PRINT "YOU FOUND THE TREASURE IN"; T: GO TO S50
480 NEXT T

490 PRINT "YOU FELL INTO THE OCEAN."
500 IF RND(1)<S THEN PRINT "EATEN BY SHARKS!": GO TO S50
510 S=S+.S: R=RT: C=CT: IF S>1 THEN S=1
520 PRINT "THE PROBABILITY OF BEING EATEN"
530 PRINT "BY A SHARK NEXT TIME IS"; S; "."
540 GO TO 480

S50 INPUT "PLAY AGAIN"; Y$
S60 IF Y$="Y" THEN RUN
S70 END

580 DATA 2,3,3,S,3,9,4,1,7,2,8,B
590 DATA 1,2,3,7,S,2,6,B,B,3,B,6

```

LOST TREASURE MODIFICATIONS

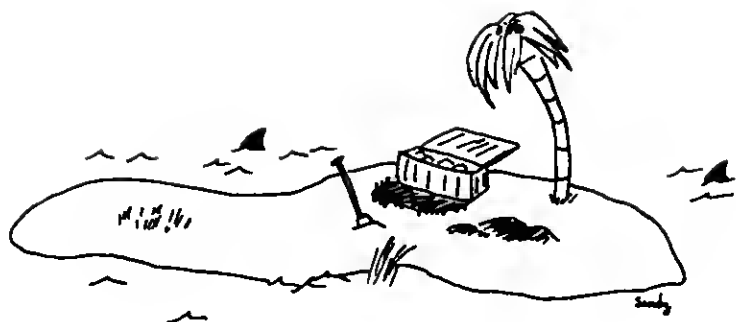
Minor

1. Probability of first shark attack -- line 20
2. Grid size -- lines 30, 180, 190, 460
3. Number of woods -- lines 60, 580
4. Number of mountains -- lines 100, 590
5. Landmarks' locations -- lines 140, 150, 160
6. Location of the treasure -- line 170
7. Movement error -- lines 380, 420
8. Amount you disturb shark -- line 510

Major

1. Vary number and amount of treasure.
2. Add parameters of water and/or food to maintain your energy level.
3. Hunt a moving treasure.
4. Modify direction of movement.
5. Add quicksand.
6. Include landmarks placed at random that are not on the map.
7. Randomly place treasure before each hunt.

NOTES



GONE FISHING

You are going on a fishing trip. The sea is an 8X8 grid, forming 64 fishing locations. You will start at the dock, square (1,1), and try to catch as many pounds of fish as you can. You may move one square at a time horizontally or vertically by entering a north(N), south(S), east(E), or west(W). Entering an F allows you to fish in the same place again, and a B allows you to start another fishing trip immediately. If you select a direction that takes you off the grid, your ship will sink. You must return to the dock in sixty moves, which is equivalent to six hours. If you don't return in time, half of your catch will spoil.

The chance of catching fish is different for each square and is determined at the beginning of the trip. The chance of catching fish in a given square will remain the same throughout the trip or will decrease if the fish are scared by a shark. The maximum number of fish that can be caught in each square (density) is also determined at the beginning of the simulation. This number varies from 1 to 5. The maximum number of fish you can catch in a square will decrease only if sea gulls eat some of the bait. The maximum weight of a fish in a particular square is the product of the row and column; therefore, the further out you go, the bigger the fish.

The longer you fish, the greater the chance of an afternoon storm occurring. If you hit a storm, you will lose .5 hour. One of the more difficult maneuvers of the trip is to fish as long as necessary to accumulate a large catch without getting lost in a storm. Also, there is a 4% chance that you will experience some unexpected event during each move of the trip. Be sure you return to the dock before six hours have elapsed. Your rating as a fisherman will be the number of pounds of fish you catch divided by five.

You may wish to use the fishing grid on page 4.6 to record the best fishing spots. A small marker can be used to keep track of your location on the grid.



Sample Run

RUN

NO BITES

AT LOCATION 1 1

TOTAL LBS. THIS TRIP IS 0.

YOU HAVE FISHEO FOR 0 HOURS.

MOVE(N,S,E,W,F,B)? E

NO BITES

AT LOCATION 1 2

TOTAL LBS. THIS TRIP IS 0.

YOU HAVE FISHEO FOR .1 HOURS.

MOVE(N,S,E,W,F,B)? S

YOU CAUGHT 1 FISH,

EACH WEIGHING 2 LBS.

AT LOCATION 2 2

TOTAL LBS. THIS TRIP IS 2.

YOU HAVE FISHEO FOR .2 HOURS.

MOVE(N,S,E,W,F,B)? S

NO BITES

AT LOCATION 3 2

TOTAL LBS. THIS TRIP IS 2.

YOU HAVE FISHEO FOR .3 HOURS.

MOVE(N,S,E,W,F,B)? E

YOU CAUGHT 4 FISH,

EACH WEIGHING 2 LBS.

AT LOCATION 3 3

TOTAL LBS. THIS TRIP IS 10.

YOU HAVE FISHEO FOR .4 HOURS.

MOVE(N,S,E,W,F,B)? E

NO BITES

AT LOCATION 4 6

TOTAL LBS. THIS TRIP IS 10.

SEA GULLS ATE SOME OF YOUR BAIT.

CATCH WILL BE SMALLER THIS TRIP.

YOU HAVE FISHEO FOR .8 HOURS.

MOVE(N,S,E,W,F,B)? S

YOU CAUGHT 4 FISH,
EACH WEIGHING 15 LBS.

AT LOCATION 4 B

TOTAL LBS. THIS TRIP IS 155.

YOU CAUGHT A 50 LB. SHARK.

TOTAL LBS. THIS TRIP IS 205.

YOU HAVE FISHEO FOR 1.8 HOURS.

MOVE(N,S,E,W,F,B)? W

YOU CAUGHT 1 FISH,
EACH WEIGHING 3 LBS.

AT LOCATION 3 3

TOTAL LBS. THIS TRIP IS 208.

WATER SPOUT DISPLACES YOU.

YOU ARE NOW AT LOCATION 4 5

YOU HAVE FISHEO FOR 2.6 HOURS.

MOVE(N,S,E,W,F,B)? W

NO BITES

AT LOCATION 1 2

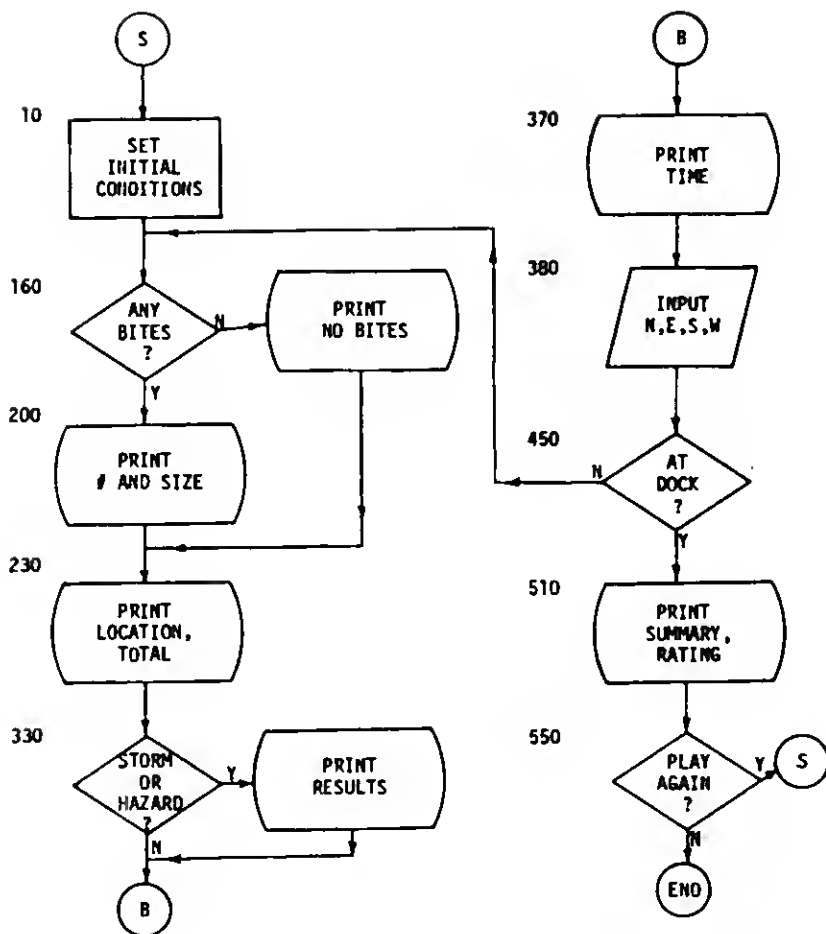
TOTAL LBS. THIS TRIP IS 211.

YOU HAVE FISHEO FOR 3.2 HOURS.

MOVE(N,S,E,W,F,B)? W

YOU ARE BACK AT THE DOCK
AFTER 3.2 HOURS OF FISHING
CLEAN 211 LBS. OF FISH.
YOU RATE 42 AS A FISHERMAN.

GONE FISHING FLOWCHART



GONE FISHING PROGRAM

Variables

P(I,J) The probability of catching a fish
 O(I,J) The maximum number of fish in square (I,J), from 1 to 8
 W Weight of each fish caught, from 1 to RXC
 P The total number of pounds of fish caught at a given time
 R Row in which you are fishing
 C Column in which you are fishing
 N Number of fish caught in a given turn
 T Time in tenths of an hour, maximum 6 hours
 M\$ Move(N,E,S,W,F,B), where N,E,S, and W are directions, F allows you to fish again in the same square, and B allows you to start the fishing trip over again

Listing

```

S      REM SET PROBABILITIES AND DENSITY
10     DIM P(8,8),O(8,8)
20     FOR I=1 TO 8: FOR J=1 TO 8
30       P(I,J)=.7*RND(1)
40       O(I,J)=INT(RND(1)*5+1)
50     NEXT J,I
60     P(1,1)=0: P=0: R=1: C=1

145    REM MAIN LOOP
150    FOR T=0 TO 6 STEP .1
160      IF RND(1)>P(R,C) OR O(R,C)<1 THEN PRINT "NO BITES": GO TO 220
170      N=INT(RND(1)*D(R,C)+1)
180      W=INT(RND(1)*R*C)+1
190      P=P+N*W
200      PRINT "YOU CAUGHT"; N; "FISH,"
210      PRINT "EACH WEIGHING"; W; "LBS.,"
220      PRINT "AT LOCATION"; R; C
230      PRINT "TOTAL LBS. THIS TRIP IS"; P; "."

325    REM UNEXPECTED EXPERIENCES
330    IF RND(1)<T/60 THEN PRINT "STORM -- LOST 1/2 HOUR": T=T+.5
340    J=INT(100*RND(1))+1
350    IF J>4 THEN 370
360    ON J GO SUB 600,700,800,900

370    PRINT "YOU HAVE FISHED FOR"; T; "HOURS."
380    INPUT "MOVE (N,S,E,W,F,B)"; M$
390    IF M$="E" THEN C=C+1
400    IF M$="N" THEN R=R-1
410    IF M$="W" THEN C=C-1
420    IF M$="S" THEN R=R+1
430    IF M$="B" THEN RUN
440    IF R<1 OR R>8 OR C<1 OR C>8 THEN PRINT "GROUNDED--SUNK!": GO TO 550
450    IF R=1 AND C=1 THEN GO TO 500
460    NEXT T

470    PRINT "TIME UP. THE SUN HAS SET."
480    PRINT "HALF OF YOUR CATCH HAS SPOILED."
490    P=P/2
  
```

```

495 REM SUMMARY OF TRIP
500 IF T=0 THEN PRINT "STILL AT DOCK": GO TO 10
510 PRINT "YOU ARE BACK AT THE DOCK"
520 PRINT "AFTER"; T; "HOURS OF FISHING."
530 PRINT "CLEAN"; P; "LBS. OF FISH."
540 "YOU RATE"; INT(P/S); "AS A FISHERMAN."
550 INPUT "ANOTHER FISHING TRIP(Y,N)"; X$
560 IF X$="Y" THEN RUN
570 ENO

```

```

595 REM SUBROUTINES
600 IF R+C<9 THEN RETURN
610 PRINT "FISH SCARED BY SHARK."
620 PRINT "NOT BITING AS OFTEN."
630 FOR I=1 TO 8: FOR J=1 TO 8
640 P(I,J)=P(I,J)-.1
650 NEXT J,I
660 RETURN
700 PRINT "SEA GULLS ATE SOME OF YOUR BAIT."
710 PRINT "CATCH WILL BE SMALLER THIS TRIP."
720 FOR I=1 TO 8: FOR J=1 TO 8
730 O(I,J)=D(I,J)-1
740 NEXT J,I
750 RETURN
800 PRINT "WATER SPOUT DISPLACES YOU."
810 R=INT(B*RNO(1)+1)
820 C=INT(B*RNO(1)+1)
830 PRINT "YOU ARE NOW AT LOCATION"; R; C
840 T=T+.2
850 RETURN
900 PRINT "YOU CAUGHT A 50 LB. SHARK."
910 P=P+50
920 PRINT "TOTAL LBS. THIS TRIP IS"; P; "."
930 RETURN

```

GONE FISHING MODIFICATIONS

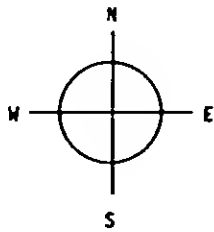
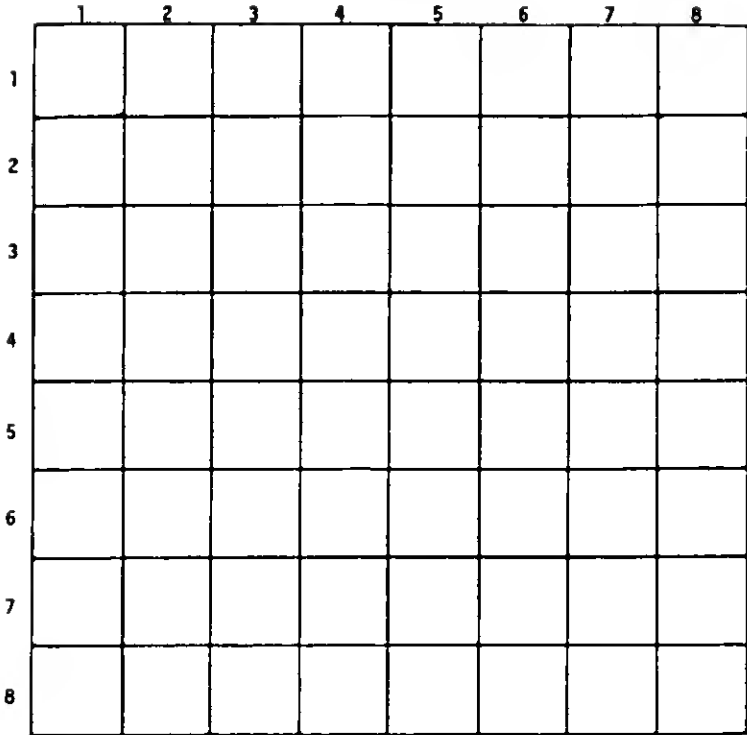
Minor

1. Grid size -- lines 10, 20, 440, 630, 720, 810, and 820
2. Maximum probability of catching fish in a square -- line 30
3. Maximum density of fish in a square -- line 40
4. Maximum time of fishing -- line 150
5. Storm probability -- line 330
6. Rating scale -- line 540

Major

1. Catch different kinds of fish, such as, sharks, whales, or mermaids.
2. Change the goal to catching the biggest fish.
3. Use fuel to run the boat.
4. Add a choice of hook sizes and fishing depth.
5. Add different kinds of hazards, such as whales, reefs, UFO's.
6. Let fishing success depend on time of day.
7. Fix weather conditions and fishing conditions at the beginning of the trip.
8. Utilize sonar devices to help locate fish.
9. Allow ship to move in a diagonal direction.

FISHING MAP



SPACE FLIGHT

Scenario

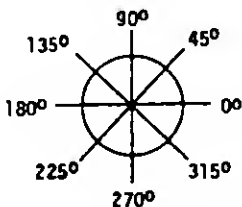
In this simulation, you are living in the year 2062 as the captain of a space ship. Your orders are to deliver medical supplies from Alpha at coordinates (10,10) to Beta at coordinates (80,80). Your rating as a space pilot will depend upon how fast you can make the trip.

During each time interval, you will be able to determine the following information:

1. Total time elapsed
2. Location in terms of X and Y coordinates
3. Amount of fuel left
4. Speed
5. The angle at which you are moving
6. Your distance from the planet.

To change direction or to increase or decrease speed, you can fire one of two kinds of rockets: main (M) and half (H). These rockets take one unit and 1/2 unit of fuel, respectively. A "C" will allow you to coast for five time intervals.

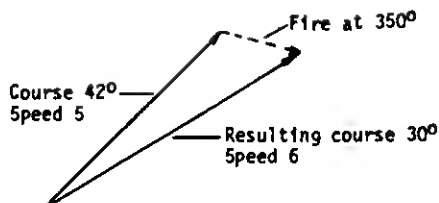
Once you decide how much fuel you are going to burn, you must decide on the direction in which you will be firing the rockets. You are able to rotate your space ship with small thrusters as it drifts in space. The directions are shown below:



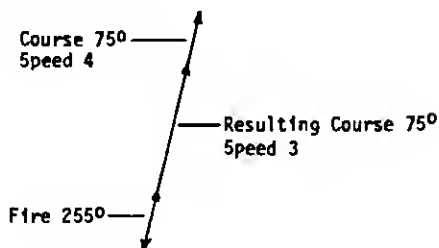
Once you fire your main rocket for three or four turns to increase your speed, you can conserve fuel by drifting through space. You must start to fire in the opposite direction to slow down before arriving at Beta. In order to meet arrival conditions, you must be within a distance of one and at a speed of less than one.

You may wish to make copies of the grid at the end of this section to aid in plotting your course. If you find that you are off course, you may have to fire a "correction" rocket. In order to estimate the angle of firing, you can use a force diagram as shown below.

Example 1: Correction



Example 2: Retrofire



Sample Run

```

DATA READOUT
0 HOURS 10 LITERS
LOCATION 10 10
VELOCITY: 0
DEGREES: 0
D=98.995
COMMANO(O,M,H,C)? M
ANGLE? 45

```

```

DATA READOUT
.01 HOURS 9 LITERS
LOCATION 10.6776 10.67
VELOCITY: .952905
DEGREES: 45
D=98.942

```

.
.
.

```

DATA READOUT
.05 HOURS 5 LITERS
LOCATION 20.1487 20.8211
VELOCITY: 5.0035
DEGREES: 50
D=84.1685
PROBLEM SUPPORT SYSTEM
COMMANO(O,M,H,C)? 0

```

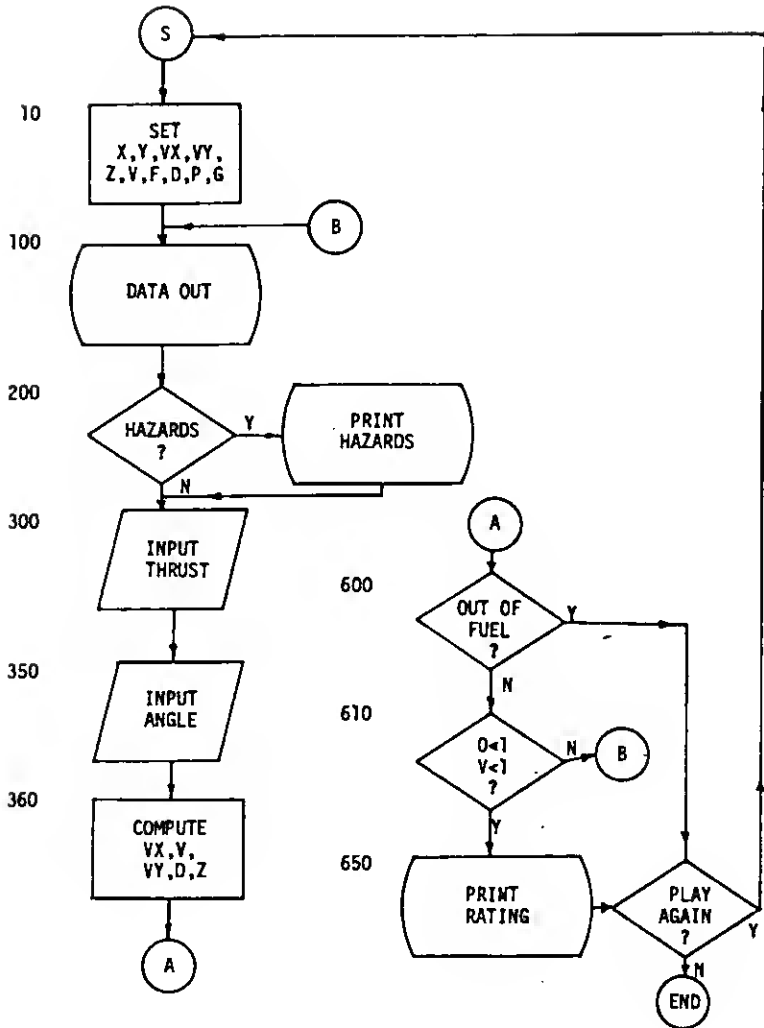
.
.

```

DATA READOUT
.33 HOURS 1 LITERS
LOCATION 79.1844 81.0019
VELOCITY: .023181
DEGREES: 58
D=1.29189
COMMANO(O,M,H,C)? H
ANGLE? 315
ARRIVED!
THE TRIP TOOK .33 HOURS.
YOUR RATING IS 66.
PLAY AGAIN? N
OK

```

SPACE FLIGHT FLOWCHART



SPACE FLIGHT PROGRAM

Variables

X,Y Location
 VX,VY Speed
 Z Angle of coast
 V Velocity
 T Time
 O Distance to planet
 J Index for hazards
 F Fuel
 A Angle input
 L,M Temporary Variables
 R Rating
 F1 Coast count
 G Accuracy of gyros



Listing

```

10  X=10: Y=10: VX=0: VY=0: Z=0: V=0
20  F=10: D=98.995: P=3.1416: G=1
30  FOR T=0 TO 10 STEP .01

100  PRINT "    DATA READOUT:"; ?
110  PRINT T; "HOURS    "; F; "LITERS"
120  PRINT "LOCATION:"; X; Y: PRINT "VELOCITY:"; V
130  PRINT Z; "DEGREES"
140  PRINT "DISTANCE:"; O

200  J=INT(50*RND(1)+1)
210  IF J<6 THEN PRINT "PROBLEMS: ";
220  ON J GO SUB 230,240,250,260,270: GO TO 290
230  PRINT "GYROS  ANGLE ERROR": G=G+1: RETURN
240  PRINT "FUEL LINE": F=F-.5: RETURN
250  PRINT "LIFE SUPPORT": T=T+.05: RETURN
260  PRINT "ALIENS": VX=D: VY=D: RETURN
270  PRINT "METEORS.": VX=VX+RND(1)-.5: VY=VY+RND(1)-.5
280  RETURN

290  IF F1>0 THEN F1=F1-1: GO TO 450
300  INPUT "COMMAND(O,M,H,C)"; C$
310  IF C$="M" THEN B=1: GO TO 350
320  IF C$="H" THEN B=2: GO TO 350
330  IF C$="C" THEN F1=5
340  GO TO 450
350  INPUT "ANGLE"; A: A=A+(20*G*RND(1)-10*G)
360  A=A*P/180
370  L=COS(A): M=SIN(A): F=F-1/B
380  VX=VX+(1+.4*RND(1)-.2)*L/B
390  VY=VY+(1+.4*RND(1)-.2)*M/B
400  IF VX=D AND VY>=0 THEN Z=90: GO TO 450
410  IF VX=0 AND VY<0 THEN Z=270: GO TO 450
420  Z=ATN(VY/VX): Z=Z*180/P
430  Z=Z+INT(10*RND(1)): Z=INT(Z)
440  IF VX<0 THEN Z=Z+180
450  X=X+VX: Y=Y+VY
  
```



```

530  V=SQR(VX+2+VY+2)
540  D=SQR((X-80)+2+(Y-80)+2)

600  IF F<0 THEN PRINT "OUT OF FUEL": GO TO 660
610  IF D<1 AND V<1 THEN PRINT "ARRIVED": GO TO 630
620  NEXT T
630  PRINT "THE TRIP TOOK"; T; "HOURS."
640  R=200*T
650  PRINT "YOUR RATING IS"; R; "."
660  INPUT "PLAY AGAIN"; Y$
670  IF Y$="Y" THEN RUN
680  END

```

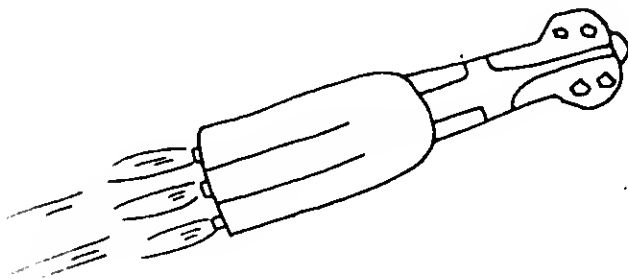
SPACE FLIGHT MODIFICATIONS

Minor

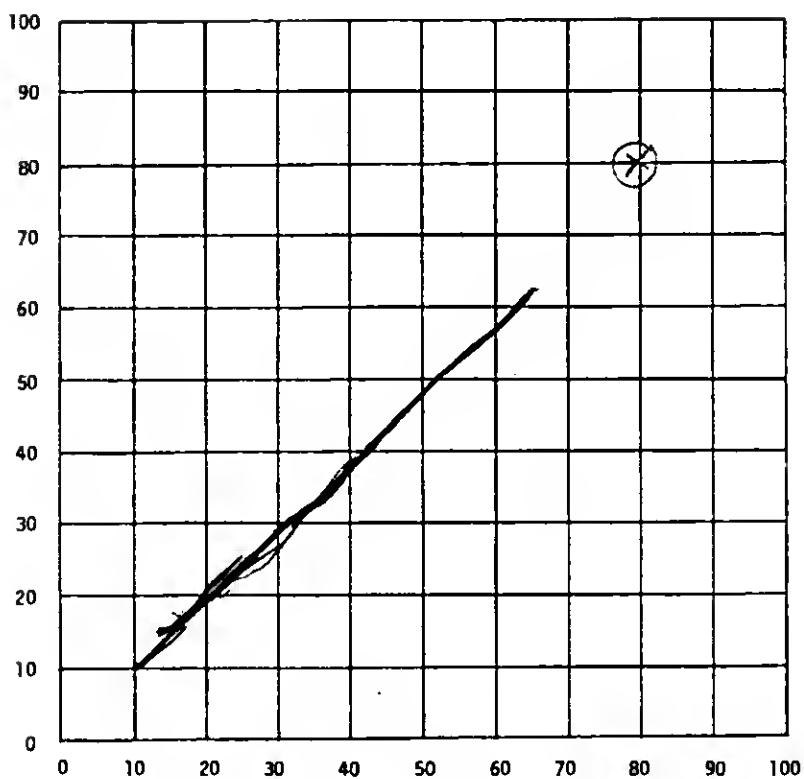
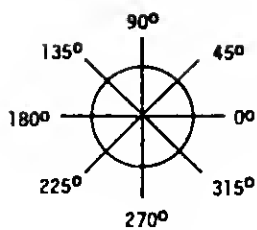
1. Starting position -- lines 10,20
2. Amount of fuel -- line 20
3. Time limit -- line 30
4. Planets location -- lines 540, 20
5. Arrival conditions -- line 610
6. Probability of problems -- line 200

Major

1. One must fire small thruster rockets to rotate ship.
2. Have meteors hit ship.
3. Use meteor shields.
4. Fight aliens.
5. Visit more than one planet.
6. Provide planets with gravitational force.
7. Have refueling stations.



Sunday



FOREST FIRE

Scenario

A lightening storm has ignited fires in a forest. Your task is to put out the fires and save as many trees as possible. The forest is divided into 81 sectors formed by a 9X9 grid. Each sector is identified by the number of its row and column. The symbol, ".", represents woods, an "*" represents fire, and a blank space represents burnt out woods.

The chance of an existing fire spreading to adjacent wooded areas is 70%. Fires last for nine turns before burning out.

You have two weapons with which to fight the fire. You can drop chemicals that are designed to extinguish the fires in a specified sector. The chance that the drop will affect the fires in this sector and its eight adjacent sectors is 50%. For example, if there are six fires burning in a nine-square area, approximately three will be affected by the chemicals. The effect of chemicals is to reduce the number of turns before the fire burns out by three. Since a fire lasts only nine turns, three successful chemical hits will be needed to extinguish a fire. If the fire has been burning for six turns, then one hit will suffice.

The second weapon available to you is a backfire. To start a backfire, you must respond to the row input with a zero. You will then be asked for a backfire row and column. The sector in which a backfire is started must be wooded. This backfire will not spread and will burn out in the next turn, forming a barrier against the spread of fire.

Your rating will be the number of trees remaining after all the fires are out, plus 30.



Sample Run

#1

	1	2	3	4	5	6	7	8	9
1
2
3
4	*	.	.
5	*	.	.
6	.	.	*
7
8
9

ROW? 0
BACKFIRE ROW? 4
BACKFIRE COLUMN? 7

#4

	1	2	3	4	5	6	7	8	9
1
2
3	*	.	.
4	*	.	.
5	.	*	.	.	.	*	*	.	.
6	.	.	*	.	.	*	*	.	.
7	.	*
8
9

ROW? 6
COLUMN? 3

#12

	1	2	3	4	5	6	7	8	9
1	.	.	.	*	*	.	*	.	.
2	.	.	.	*	*
3
4
5	.	*
6
7	*	.	.	.
8	.	.	.	*	*
9	.	.	*	.	*

ROW? 8
COLUMN? 7

#2

	1	2	3	4	5	6	7	8	9
1
2
3
4	*	*	.	.	.
5	*
6	.	.	*
7	.	*
8
9

ROW? 0
BACKFIRE ROW? 5
BACKFIRE COLUMN? 7

#11

	1	2	3	4	5	6	7	8	9
1	.	.	.	*	*	*	*	.	.
2	.	.	.	*	*	*	.	.	.
3
4
5	*
6	*
7	*	.	.	.	*
8	*
9	.	*	.	*

ROW? 6
COLUMN? 2

#16

	1	2	3	4	5	6	7	8	9
1	.	.	.	*
2
3
4
5
6	.	.	.	*
7
8
9

ROW? 6
COLUMN? 6

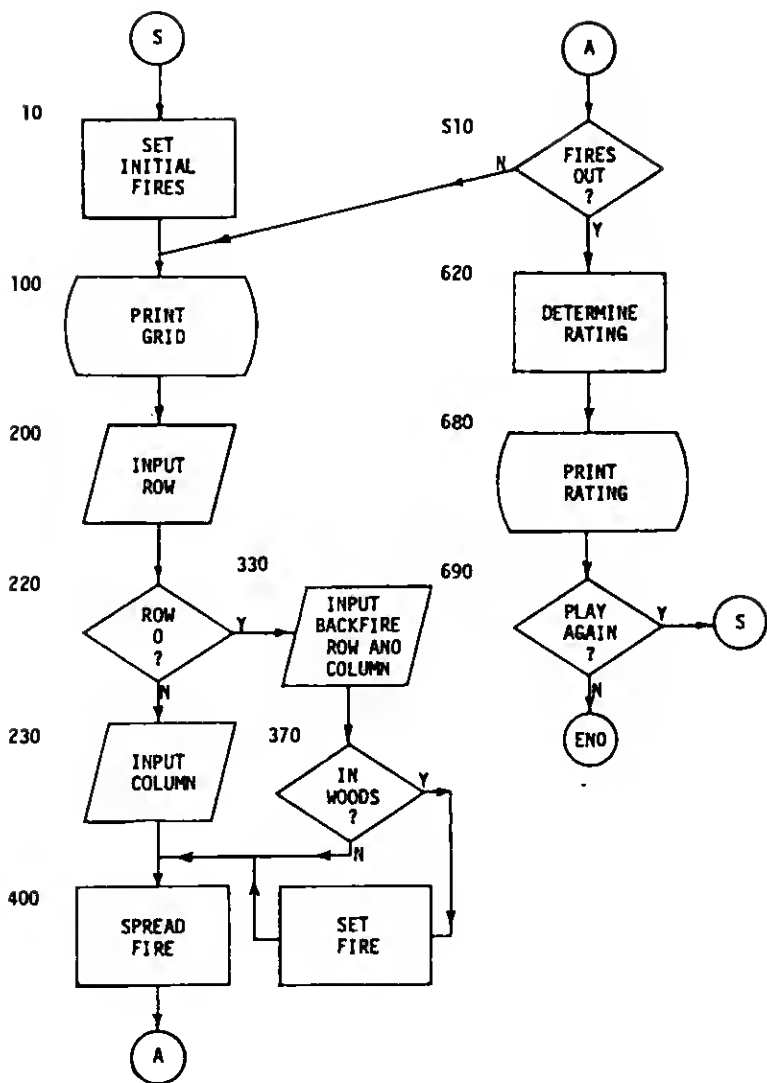
#3

	1	2	3	4	5	6	7	8	9
1
2
3	*	.	.
4	*	.	.	.
5	.	*	.	.	*	*	.	.	.
6	.	*	*	.	*
7	*
8
9

ROW? 0
BACKFIRE ROW? 6
BACKFIRE COLUMN? 7

YOUR RATING IS 69.
PLAY AGAIN?

FOREST FIRE FLOWCHART



FOREST FIRE PROGRAM

Variables

L(R,C) Burnt woods: 0, fire: 1-9, woods: 10, temporary variable: 11
 R Row
 C Column
 I Row number increment
 J Column number increment
 A Adjacent row
 B Adjacent column
 F Count
 T Temporary variable
 R Rating

Listing

```

10 DIM L(9,9)
20 FOR R=1 TO 9: FOR C=1 TO 9
30 L(R,C)=10
40 NEXT C,R
50 FOR I=1 TO 3
60 R=INT(9*RND(1)+1)
70 C=INT(9*RND(1)+1)
80 L(R,C)=9
90 NEXT I

95 REM PRINT GRID
100 PRINT " 1 2 3 4 5 6 7 8 9"
110 FOR R=1 TO 9
120 PRINT R; " ";
130 FOR C=1 TO 9
140 IF L(R,C)=10 THEN PRINT "."; GO TO 170
150 IF L(R,C)>0 AND L(R,C)<10 THEN PRINT "*"; GO TO 170
160 PRINT " ";
170 NEXT C
180 PRINT: NEXT R

195 REM INPUT ROUTINE
200 INPUT "ROW"; R
210 IF R<0 OR R>9 THEN 200
220 IF R=0 THEN 330
230 INPUT "COLUMN"; C
240 IF C<1 OR C>9 THEN 230
250 FOR I=-1 TO 1: FOR J=-1 TO 1
260 A=R+I: B=C+J
270 IF A<1 OR A>9 OR B<1 OR B>9 THEN 310
280 IF L(A,B)<1 OR L(A,B)=10 THEN 310
290 IF RND(1)>.5 THEN 310
300 L(A,B)=L(A,B)-3
310 NEXT J,I
320 GO TO 400

330 INPUT "BACKFIRE ROW"; R
340 IF R<1 OR R>9 THEN 330
350 INPUT "BACKFIRE COLUMN"; C
360 IF C<1 OR C>9 THEN 350
  
```

```

370 IF L(R,C)=10 THEN L(R,C)=2

395 REM SPREAD FIRE
400 FOR R=1 TO 9: FOR C=1 TO 9
410 IF L(R,C)<1 OR L(R,C)>9 THEN 500
420 IF L(R,C)<3 THEN 500
430 I=INT(3*RND(1)-1)
440 J=INT(3*RND(1)-1)
450 A=R+I: B=C+J
460 IF A<1 OR A>9 OR B<1 OR B>9 THEN 500
470 IF L(A,B)<>10 THEN 500
480 IF RND(1)<.3 THEN 500
490 L(A,B)=11
500 NEXT C,R

505 REM BURN FIRE AND COUNT
510 F=0
520 FOR R=1 TO 9
530 FOR C=1 TO 9
540 T=L(R,C)
550 IF T=11 THEN T=9
560 IF T>0 AND T<10 THEN T=T-1: F=F+1
570 L(R,C)=T
580 NEXT C,R
590 IF F<1 THEN 620
600 GO TO 100

615 REM COUNT WOODS RATING
620 C=0
630 FOR R=1 TO 9: FOR C=1 TO 9
640 IF L(R,C)=10 THEN W=W+1
650 NEXT C,R
660 R=W*30
670 IF R>100 THEN R=100
680 PRINT "YOUR RATING IS"; R; "."
690 INPUT "PLAY AGAIN"; Y$
700 IF Y$="Y" THEN RUN
710 END

```

FOREST FIRE MODIFICATIONS

Minor

1. Number of beginning fires -- line 50
2. Location of beginning fires -- lines 60, 70
3. Probability of putting out fire -- line 290
4. Amount fire burns out each turn -- line 300
5. Size of backfire -- line 370
6. Probability of spread -- line 480
7. Size of spread fires -- line 550
8. Rating scale - lines 660, 670

Major

1. Change grid size.
2. Randomly choose location of beginning fires.
3. Add time to move from one place to another.
4. Have wind speed and direction affect the spread of the fire.
5. Include barriers such as lakes and roads.
6. Have some of the sectors burn faster than others.

NAUTICAL NAVIGATION

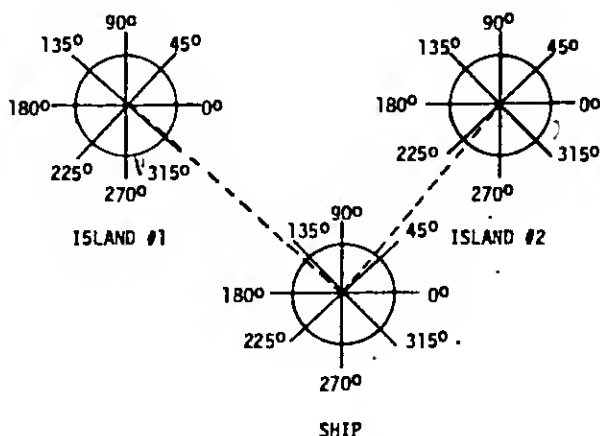
Scenario

Your task is to navigate a sailboat that has an electronic direction finder to three different islands in the South Pacific. You do not have to dock at the islands, but only come close enough to make a visual sighting. The minimum sighting distance will vary from five to ten miles, depending upon weather conditions.

The islands are located at coordinates (200,300), (600,300), and (300,100). Your starting location will be approximately (200,200). You will need graph paper and an inexpensive protractor and ruler in order to plot your course.

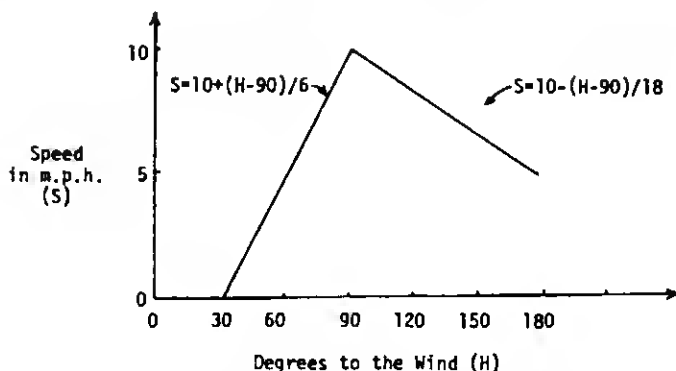
Each turn you will receive information about your bearings in degrees from each of the three islands. For convenience, you will also receive the bearings from the ship to each of the islands. The example below shows how the bearings are determined. If you know the bearing from two of the three islands, you can locate the ship; however, there are some random errors in the readings, so it might be wise to use the readings from all three islands.

Bearing from island #1: 317° ; bearing to island #1: 138° .
Bearing from island #2: 230° ; bearing to island #2: 50° .



After you locate your position, you must determine your heading and the length of time you wish to remain on this course. You can use the heading from the ship to the island of your destination to determine the ship's heading. Since you are in a sailboat, your speed will depend on your direction with respect to an easterly wind. In order to make any progress toward the East, you must tack at either 45° or 315° . The speed

of the sailboat as a function of its direction is shown in the graph below.



The fastest speed of ten miles per hour is achieved when the boat is perpendicular to the wind -- heading either directly north (90°) or south (270°). When the boat is running with the wind directly behind it, its speed is about half the maximum speed or five m.p.h.

Once you determine the heading, you must determine the length of time you wish to remain on the heading or the length of time you wish to travel before the next navigational check. The speed at 70° is about 6.7 m.p.h. In ten hours, you would travel about 67 miles. Of course, the wind speed varies; so you may wish to make one or two navigational checks on a long run.

You can visit the three islands in any order. You must compute the angle and time so the end of a run is within five to ten miles of an island. Since visibility conditions vary, you may have to wait for a turn to allow sighting conditions to improve.

Your rating as a navigator will depend on the number of navigational checks required and the amount of time for the trip. A good sailor should be able to complete the trip with a rating close to 100.

Sample Run

NAVIGATION CHECK 1
BEARING FROM 1: 279 TO: 99
BEARING FROM 2: 197 TO: 17
BEARING FROM 3: 136 TO: 316
ELAPSED TIME 0
HEADING? 99
TIME? 33

NAVIGATION CHECK 2
BEARING FROM 1: 97 TO: 277
BEARING FROM 2: 158 TO: 338
BEARING FROM 3: 108 TO: 288
ELAPSED TIME 32.9694
HEADING? 277
TIME? 20

NAVIGATION CHECK 3
VISITED 1
BEARING FROM 1: 84 TO: 264
BEARING FROM 2: 179 TO: 359
BEARING FROM 3: 115 TO: 295
ELAPSED TIME 52.9576
HEADING? 295
TIME? 30

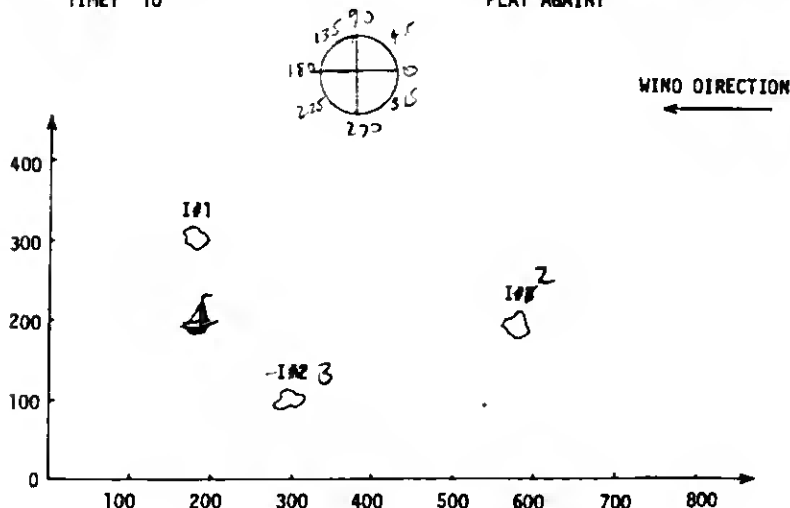
NAVIGATION CHECK 4
VISITED 1
BEARING FROM 1: 296 TO: 116
BEARING FROM 2: 201 TO: 21
BEARING FROM 3: 117 TO: 297
ELAPSED TIME 82.9246
HEADING? 297
TIME? 10

NAVIGATION CHECK 5
VISITED 1
BEARING FROM 1: 296 TO: 116
BEARING FROM 2: 209 TO: 29
BEARING FROM 3: 114 TO: 294
ELAPSED TIME 92.8834
HEADING? 294
TIME? 3

NAVIGATION CHECK 6
VISITED 1
VISITED 3
BEARING FROM 1: 296 TO: 116
BEARING FROM 2: 212 TO: 32
BEARING FROM 3: 119 TO: 299
ELAPSED TIME 95.8568
HEADING? 60
TIME? 120

NAVIGATION CHECK 7
VISITED 1
VISITED 3
BEARING FROM 1: 35 TO: 215
BEARING FROM 2: 92 TO: 272
BEARING FROM 3: 58 TO: 238
ELAPSED TIME 215.833
HEADING? 272
TIME? 28

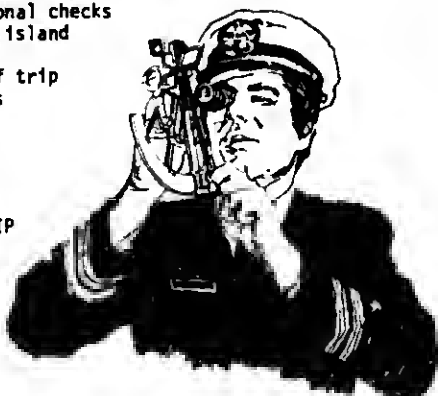
TRIP COMPLETED IN 243.859 HOURS
NUMBER OF NAVIGATIONAL CHECKS 7
YOUR RATING IS 66
PLAY AGAIN?



NAUTICAL NAVIGATION PROGRAM

Variables

O(3)	Set to 1 if arrived at destination
A(3),B(3)	Coordinates of islands
X,Y	Coordinates of ship
E	Total elapsed time
C	Number of navigational checks
L	Angle bearing from island
H	Heading of ship
T	Time for one leg of trip
A,B	Temporary variables
Y\$	Play again



Listing

```

5    REM PLACE ISLANDS AND SHIP
10   OIH A(3), B(3), O(3)
20   E=0: P=3.14159
30   FOR I=1 TO 3
40   READ A,B
50   A(I)=10*A: B(I)=10*B
60   O(I)=0
70   NEXT I
80   DATA 20,30,60,30,30,10
90   X=175+S0*RNO(1): Y=175+S0*RNO(1)

95   REM START MAIN LOOP
100  FOR C=1 TO 100
110  PRINT "NAVIGATION CHECK"; C
120  FOR I=1 TO 3
130  IF O(I)=1 THEN PRINT "VISITEO"; I
140  NEXT I

150  FOR I=1 TO 3
160  A=A(I): B=B(I)
170  GO SUB 600: L=L+2.S-S*RNO(1)
180  L=L+180: IF L>360 THEN L=L-360
190  PRINT "BEARING FROM"; I, "IS"; INT(L);
200  IF L>=180 THEN L=L-180: PRINT " TO"; INT(L): GO TO 220
210  IF L<180 THEN L=L+180: PRINT " TO"; INT(L)
220  NEXT I

225  REM INPUT
230  PRINT "ELAPSED TIME"; E
240  INPUT "HEADING"; H
250  H=H+S-10*RNO(1)
260  INPUT "TIME"; T: T=ABS(T)
270  CO=COS(H*P/180): SI=SIN(H*P/180)
280  IF H>180 THEN H=360-H
290  IF H<30 THEN S=0
300  IF H>=30 AND H<90 THEN S=10+(H-90)/6
310  IF H>90 THEN S=10-(H-90)/18
320  S=S+2*RNO(1)-1
330  T=T+(.1*RNO(1)-.05)
340  X=X+T*S*CO

```

```

350  Y=Y+T*S*SI
360  E=E+T

400  FOR I = 1 TO 3
410  D=SQRT((X-A(I)+2+(Y-B(I)+2)
420  IF O<S+10*RHO(I) THEN O(I)=1
430  NEXT I
440  IF O(1)+O(2)+O(3)=3 THEN GO TO 500
450  NEXT C
460  PRINT "EXCEED NAVIGATION CHECK": GO TO S30
470  PRINT "TRIP COMPLETED IN"; E; "HOURS."
480  PRINT "NUMBER OF NAVIGATION CHECKS IS"; C; "."
490  PRINT "YOUR RATING IS"; 170-(INT(E+10*C/3))
500  INPUT "PLAY AGAIN"; Y$
510  IF Y$="Y" THEN RUN
520  END

600  IF X=A AND Y>B THEN L=270: RETURN
610  IF X=A AND Y<B THEN L=90: RETURN
620  N=ABS(Y-B)/ABS(X-A)
630  L=ATN(N): L=180*L/P
640  IF X>A AND Y>=B THEN L=L+180
650  IF X<A AND Y>=B THEN L=L-180
660  IF X>A AND Y<B THEN L=L-180
670  RETURN

```

NAUTICAL NAVIGATION MODIFICATIONS

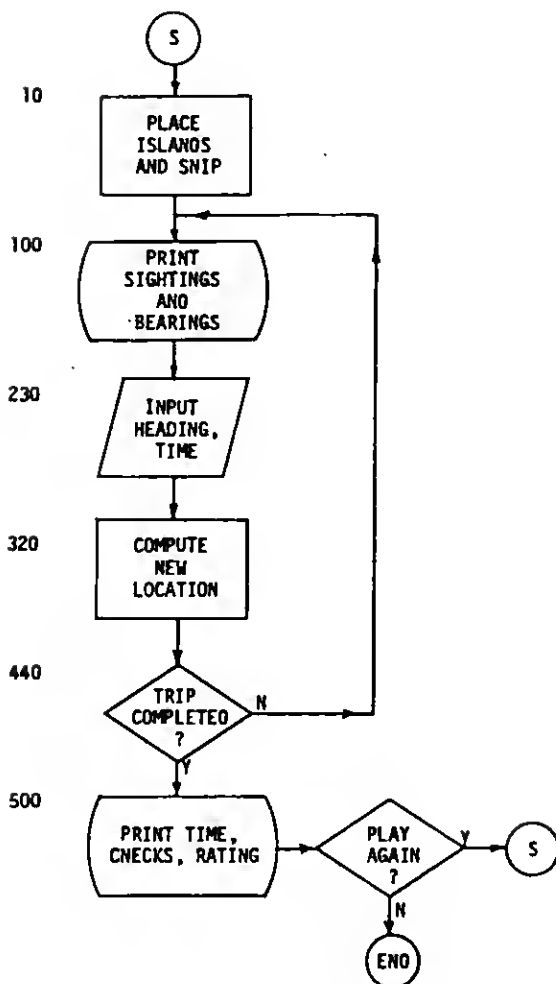
Minor

1. Location of islands -- line 80
2. Starting place of ship -- line 90
3. Error in angle -- line 170
4. Input error -- line 250
5. Speed error -- line 320
6. Time error -- line 330
7. Sighting criteria -- line 420
8. Rating -- line S20

Major

1. Change number of islands.
2. Have storms.
3. Have wind direction change.

NAUTICAL NAVIGATION FLOWCHART



BUSINESS MANAGEMENT

Scenario

In this simulation you manage a small factory that produces three different kinds of products (P1 - P3). Three different kinds of raw materials (R1 - R3) are required to produce the products. Each product requires exactly two raw materials with a different subscript. For example, to manufacture one unit of P2, you would need a unit of R1 and a unit of R3. To manufacture one unit of P3, you would need a unit of R1 and R2.

The cost of raw materials varies from \$10 to \$20 per unit. It costs from \$1 to \$9 per unit to manufacture a product from raw materials. The selling price of each finished product varies from \$50 to \$90 per unit. Prices of raw materials and manufacturing costs will vary by not more than \$2 per turn. Prices of finished products will vary by not more than \$5 per turn.

You will receive a data report at the beginning of each turn. This report will give you the number of units you have on hand, available cash, and the manufacturing costs. You can buy, manufacture, or sell each turn. In order to manufacture a given product, you must have enough of the correct kind of materials on hand.

After twelve turns (months), the materials and/or products that you have on hand will be automatically sold at the current prices and your profit will be computed.

Sample Run

ITEM	MATERIALS	PRODUCTS
1	\$0-\$16	\$0-\$72
2	\$0-\$15	\$0-\$72
3	\$0-\$17	\$0-\$73

MONTH 0 YOU HAVE \$500
 MANUFACTURING COSTS ARE \$2
 TRANSACTION 0,B,M,S? B
 AMOUNT OF MATERIALS? 10
 ITEM#? 2

ITEM	MATERIALS	PRODUCTS
1	\$0-\$16	\$0-\$67
2	\$10-\$16	\$0-\$71
3	\$0-\$16	\$0-\$73

MONTH 1 YOU HAVE \$350
 MANUFACTURING COSTS ARE \$1
 TRANSACTION 0,B,M,S? B
 AMOUNT OF MATERIALS? 10
 ITEM#? 1

ITEM	MATERIALS	PRODUCTS
1	\$10-\$18	\$0-\$63
2	\$10-\$17	\$0-\$70
3	\$0-\$18	\$0-\$68

MONTH 2 YOU HAVE \$190
 MANUFACTURING COSTS ARE \$2
 TRANSACTION O,B,M,S? M
 MANUFACTURE AMOUNT? 10
 ITEM#? 3

ITEM	MATERIALS	PRODUCTS
1	\$0-\$19	\$0-\$67
2	\$0-\$15	\$0-\$72
3	\$0-\$18	\$10-\$73

MONTH 3 YOU HAVE \$170
 MANUFACTURING COSTS ARE \$2
 TRANSACTION O,B,M,S? S
 AMOUNT TO SELL? 10
 ITEM#? 3

ITEM	MATERIALS	PRODUCTS
1	\$0-\$17	\$0-\$72
2	\$0-\$17	\$0-\$76
3	\$0-\$18	\$0-\$77

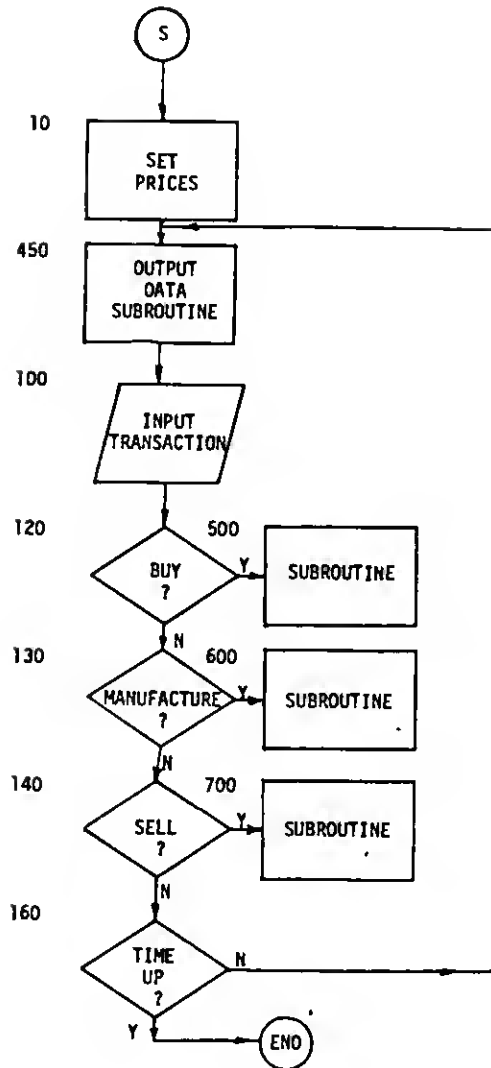
MONTH 4 YOU HAVE \$900
 MANUFACTURING COSTS ARE \$3
 TRANSACTION O,B,M,S?

.

ITEM	MATERIALS	PRODUCTS
1	\$0-\$18	\$0-\$71
2	\$0-\$12	\$0-\$62
3	\$0-\$10	\$0-\$68

MONTH 12 YOU HAVE \$2380
 MANUFACTURING COSTS ARE \$8
 TRANSACTION O,B,M,S? 0
 END OF YEAR
 YOUR PROFIT IS 1880.
 PLAY AGAIN?

BUSINESS MANAGEMENT FLOWCHART



BUSINESS MANAGEMENT PROGRAM

Variables

R(I) Number of raw materials
 C(I) Cost of one unit of raw material
 F(I) Number of finished products
 P(I) Price of one unit of finished product (\$0-\$90)
 C Cash on hand
 M Manufacturing costs (\$1-\$9) per unit
 T Time
 N Item number
 A Amount
 T\$ Input O,B,M,S

Listing

```

S      REM SET PRICES
10     DIM R(3), C(3), F(3), P(3)
20     C=500: M=2
30     FOR I=1 TO 3
40       R(I)=0: F(I)=0
50       C(I)=INT(3*RND(1)+15)
60       P(I)=INT(10*RND(1)+70)
70     NEXT I
80     FOR T=0 TO 12
90       GO SUB 450

100    PRINT "MONTH"; T; "YOU HAVE"; C: PRINT: PRINT "MANUFACTURING
      COSTS ARE $"; M
110    INPUT "TRANSACTION O,B,M,S"; T$
120    IF T$="B" THEN GO SUB 500
130    IF T$="M" THEN GO SUB 600
140    IF T$="S" THEN GO SUB 700
150    GO SUB 300
160    NEXT T

165    REM SUMMARY
170    PRINT "END OF YEAR"
180    FOR I=1 TO 3
190      C=C+R(I)*C(I)
200      C=C+F(I)*P(I)
210    NEXT I

220    C=C-500
230    PRINT "YOUR PROFIT IS"; C; "."
240    INPUT "PLAY AGAIN"; Y$
250    IF Y$="Y" THEN RUN
260    END

295    REM CHANGE PRICE SUBROUTINE
300    FOR I=1 TO 3
310      J=INT(5*RND(1)-2)
320      J=C(I)+J
330      IF J<10 OR J>20 THEN 310
340      C(I)=J
350      J=INT(11*RND(1)-5)
360      J=P(I)+J
  
```

```

370 IF J<50 OR J>90 THEN 350
380 P(I)=J
390 NEXT I

400 J=INT(5*RND(1))-2)
410 J=M+J
420 IF J<1 OR J>9 THEN 400
430 M=J
440 RETURN

445 REM OUTPUT DATA
450 PRINT "ITEM      MATERIALS      PRODUCT": PRINT
460 FOR I=1 TO 3
470 PRINT I; " "; R(I); " $"; C(I); " "; F(I); " $"; P(I):PRINT
480 NEXT I
490 RETURN

495 REM BUY MATERIALS
500 INPUT "AMOUNT OF MATERIALS"; A
510 INPUT "ITEM#"; N
520 IF N<1 OR N>3 THEN PRINT "ERROR": RETURN
530 C=C+A*C(N)
540 IF C<0 THEN 570
550 R(N)=R(N)+A
560 RETURN
570 C=C+A*C(N)
580 PRINT "INSUFFICIENT FUNOS"
590 RETURN

595 REM MANUFACTURE
600 INPUT "MANUFACTURE AMOUNT"; A: INPUT "ITEM#"; N
610 IF N<0 OR N>3 THEN PRINT "ERROR": RETURN
620 C=C-A*M
630 IF C<0 THEN PRINT "INSUFFICIENT FUNOS": C=C+A*M: RETURN

640 FOR I=1 TO 3
650 IF I=N THEN 680
660 R(I)=R(I)-A
670 IF R(I)<0 THEN PRINT "MATERIALS GONE": R(I)=R(I)+A: C=C+A*M:
RETURN
680 NEXT I: F(N)=F(N)+A: RETURN

695 REM SELL
700 INPUT "AMOUNT TO SELL"; A: INPUT "ITEM#"; N
710 IF N<0 OR N>3 THEN PRINT "ERROR": RETURN
720 F(N)=F(N)-A
730 IF F(N)<0 THEN 760
740 C=C+A*P(N)
750 RETURN
760 F(N)=F(N)+A
770 PRINT "PRODUCTS GONE"
780 RETURN

```

BUSINESS MANAGEMENT MODIFICATIONS

Minor

1. Starting amounts -- lines 20, 50, 60
2. Number of turns -- line 80
3. Amount raw materials vary -- line 310
4. Range of raw materials -- line 330
5. Amount products vary -- line 350
6. Range of products -- line 370
7. Amount manufacturing costs vary -- line 400
8. Range of manufacturing costs -- line 420

Major

1. Increase number of raw materials and finished products.
2. Have a storage fee.
3. When you buy, prices increase.
4. When you sell, prices decrease.
5. Borrow money with interest.
6. Add random events, such as strikes, shortage of materials, fires, no demand.
7. Provide names for raw materials and products.



RARE BIRDS

Scenario

In this simulation you attempt to identify as many birds as possible in a ten hour period. First, you must choose a place to watch birds. It must be in the swamp (S), the water (W), the desert (D), or the forest (F). Then you must choose a time of day -- morning (M), or evening (E). Finally, you must choose to look up in the sky -- high (H) or on the ground -- low (L). There are sixteen different birds that can be identified. The birds are classified as small or big, yellow or blue, shortbeaked or long beaked, and female or male.

After you have selected a place to watch birds, you will receive one clue about the bird and the length of time it took you to spot it. If no bird is spotted in a two-hour period, you may try a new place. After receiving your clue, you then have an opportunity to identify the bird. You should refer to the bird watching chart to determine where the birds are seen and their specific characteristics. The birds with the larger numbers are observed more frequently.

If your first identification is not correct, you will have an opportunity to try again. Each time you try, however, one point will be subtracted from your final rating. If you identify a bird that you have identified correctly before, you will be notified of the fact and may try a new place. Your final rating is determined by multiplying ten times the number of birds identified and subtracting one for each incorrect identification.



Sample Run

PLACE S,W,O,F? S
 WHEN M,E? E
 WHERE H,L? L
 THE BIRD IS YELLOW
 TIME LAPSE: 1.28
 TOTAL TIME: 1.28
 IDENTIFY I-16? 12

NOT CORRECT IDENTIFICATION
 IDENTIFY I-16? 12
 A NEW ONE!

PLACE S,W,D,F? W
 WHEN M,E? E
 WHERE H,L? H
 THE BIRD IS BIG
 TIME LAPSE: .18
 TOTAL TIME: 1.46
 IDENTIFY I-16? 11

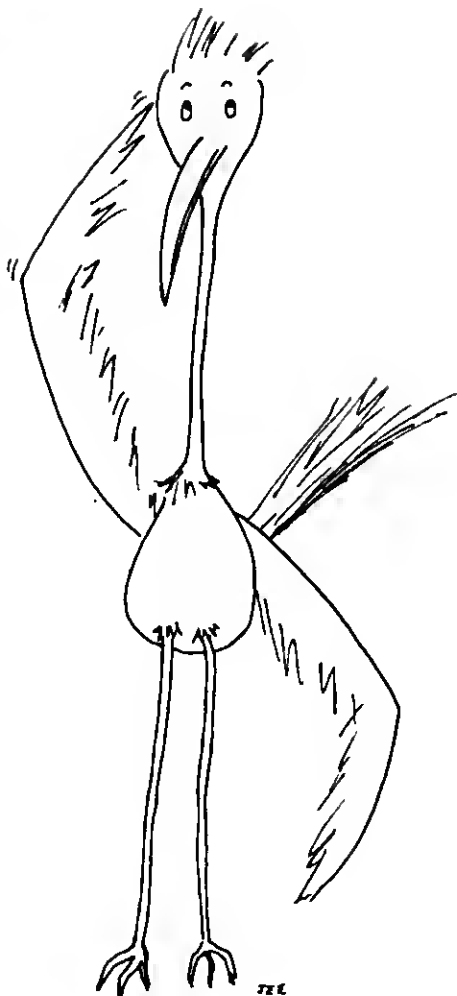
NOT CORRECT IDENTIFICATION
 IDENTIFY I-16? 9
 A NEW ONE!

PLACE S,W,O,F? S
 WHEN M,E? E
 WHERE H,L? L
 NO SIGHTINGS

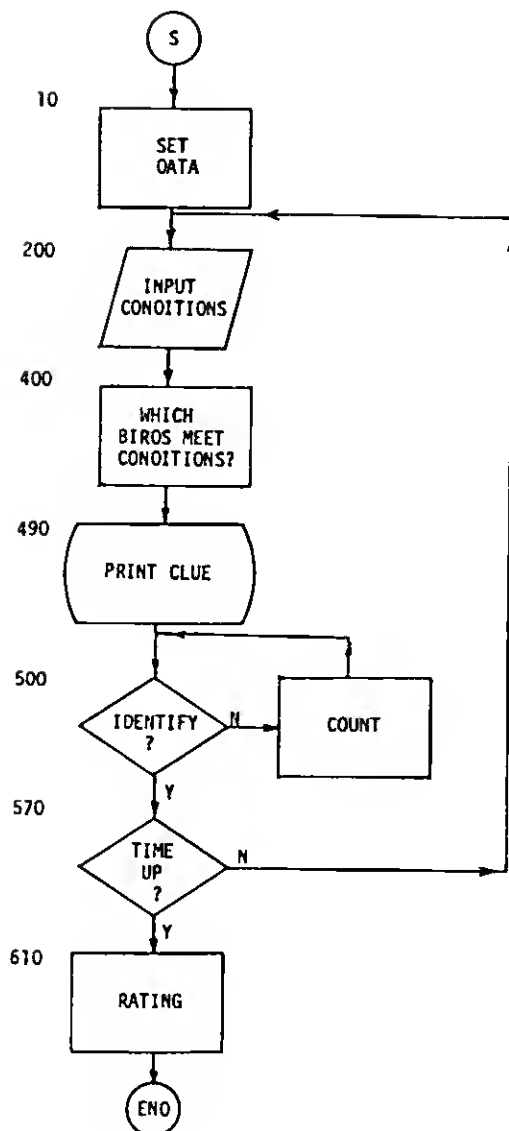
TIME UP
 YOU SAW BIRD#

1
 6
 9
 12
 15
 16

YOUR RATING IS S7
 PLAY AGAIN?



RARE BIRDS FLOWCHART



Variables

B(I,J)	I is bird (1-16); J is characteristic (1-14)
NS(I)	Name characteristic
P(I)	Probability of sighting
K,I,J,Q,N	Temporary variables
L\$	Place
T\$	When
A\$	Where
I	Lapsed time for one sighting
H	Total time
B ₁	Number of identifications
C ₁	Number of birds identified

Listing

```
5      REM SET DATA
10     H=0: DIM B(16,14), I(16), NS(8), P(16)
20     PRINT "PLEASE WAIT": FOR I=1 TO 16
30     B(I,14)=0
40     P(I)=1/(17-I)
50     READ N
60     FOR J=12 TO 1 STEP -1
70     Q=INT(N/2)
80     B(I,J)=2*(N/2-Q)
90     N=Q
100    NEXT J
110    NEXT I

120    DATA 2128, 1121, 594, 355, 3220
130    DATA 2725, 2454, 1703, 1528, 1017
140    DATA 2042, 3067, 3516, 3773, 4030, 4031

150    FOR I=1 TO 8
160    READ NS(I): NEXT I
170    DATA OIG, SMALL
180    DATA BLUE, YELLOW
190    DATA LONG BEAKED, SHORT BEAKED, FEMALE, MALE

195    REM INPUT PLACE
200    FOR I=1 TO 16: I(I)=0: NEXT
210    INPUT "PLACE S,W,O,F": L$
220    INPUT "WHEN M,E": T$
230    INPUT "WHERE H,L": A$

260    IF L$="S" THEN I(1)=1
270    IF L$="W" THEN I(2)=1
280    IF L$="O" THEN I(3)=1
290    IF L$="F" THEN I(4)=1
300    IF T$="M" THEN I(5)=1
310    IF T$="E" THEN I(6)=1
320    IF T$="S" THEN I(7)=1
330    IF T$="G" THEN I(8)=1
340    FOR I=1 TO 16: B(I,13)=0: NEXT I
350    FOR I=1 TO 16: FOR J=1 TO 8
```



```

360 IF B(I,J)<>I(J) AND B(I,J)=0 THEN 390
370 NEXT J
380 B(I,13)=1
390 NEXT I

395 REM FINO BIROS
400 FOR I=1 TO 2 STEP .02
410 J=INT(16*RND(1)+1)
420 IF B(J,13)<>1 THEN 440
430 IF RND(1)<P(J) THEN 460
440 NEXT I
450 PRINT "NO SIGHTINGS": H=H+I: GO TO 200
460 H=H+I
470 K=INT(4*RND(1)+1)
480 N=B(J,K+B)
490 PRINT "THE BIRO IS"; N$(2*K-N): PRINT "TIME LAPSE:"; I: PRINT
    "TOTAL TIME:"; H

495 REM INPUT IO
500 INPUT "IOENTIFY 1-16"; I
510 IF I=J THEN S30
520 PRINT "NOT CORRECT IOENTIFICATION": C1=C1+1: GO TO S00
530 IF B(J,14)=1 THEN PRINT "ALREADY SPOTTEO": GO TO S50
540 PRINT "A NEW ONE!": B(J,14)=1
550 IF H>10 THEN S70
560 GO TO 200

570 PRINT "TIME UP"
580 FOR I=1 TO 16
590 IF B(I,14)=1 THEN PRINT "YOU SAW BIRD #"; I: B1=B1+1
600 NEXT I
610 PRINT "YOUR RATING IS"; 10*B1-C1; ". "
620 INPUT "PLAY AGAIN"; Y$
630 IF Y$="Y" THEN RUN
640 END

```

RARE BIROS MODIFICATIONS

Minor

1. Probability of sighting -- line 40
2. Time interval per turn -- line 400
3. Total time -- line S50
4. Rating formula -- line 610

Major

1. Increase number of birds.
2. Increase characteristics of birds.
3. Allow a bird to be identified more than once.
4. Have some extremely rare birds.

Note: The birds' characteristics are stored in decimal format in statements 120, 130, and 140. Statements S0-100 convert the decimal numbers into binary and store the binary digits in B(I,J).

BIRD WATCHING CHART

B I R D	PLACE	WHEN	WHERE	S M A L L	B I G	Y E L L D W	B L U E	S B H O R K T E - D	L B D E N A K E - D	F E M A L E
1	S	E	L	S		Y		S		M
2	W	E	H	S		Y		S		F
3	D	E	L	S		Y			L	M
4	F	E	H	S		Y			L	F
5	SW	M	L	S			B	S		M
6	SD	M	H	S			B	S		F
7	S F	M	L	S			B		L	M
8	WD	M	H	S			B		L	F
9	W F	ME	HL		B	Y		S		M
10	DF	ME	HL		B	Y		S		F
11	WDF	ME	HL		B	Y			L	M
12	S DF	ME	HL		B	Y			L	F
13	SW F	M	HL		B		B	S		M
14	SWD	M	HL		B		B	S		F
15	SWDF	M	HL		B		B		L	M
16	SWDF	M	HL		B		B		L	F

DIAMOND THIEF

Scenario

An expensive diamond is stolen from a museum. Your job, as the detective assigned to the case, is to determine who stole the diamond and at what time. You deduce the solution by studying the responses made by five different suspects, one of whom is guilty. Your rating is determined by how quickly you can identify the thief.

The five suspects were wandering through a nine room museum from one p.m. to twelve midnight. They never stayed in the same room for two consecutive hours, although they may have returned to the same room more than once.

You determine who you want to question and a specific time from one to twelve. The suspect responds by giving the following information:

1. Suspect's location at specified time
2. Whether or not the diamond was seen in room #5 at the specified time
3. Who was with the suspect
4. Who the suspect saw in adjacent rooms

There is a catch, however. The innocent suspects can forget the exact room they were in and may name adjacent rooms 5% of the time instead. There is also a 5% chance that innocent people will make errors in naming people in the room with them or people whom they saw. The thief makes errors 50% of the time. Any statement made about room #5 or any statement made about the diamond is always true.

The diamond was stolen at the end of the time interval; therefore, the thief or people in room #5 with the thief will claim to have seen the diamond during the time it was stolen. Of course, after the diamond was stolen, suspects will not have seen it.

When you think you know who the thief is and the time it was stolen, then you should enter a zero in response to "suspect?". If you get either the thief or the time correct, you will get another chance, but will lose a ten question penalty on the final rating.

Sample Run

RUN
 PLEASE WAIT
 SOMEONE STOLE THE DIAMOND!!
 QUESTION 1
 SUSPECT (1-5)? 1
 TIME? 6
 SUSPECT 1 AT TIME 6
 I WAS IN ROOM 8
 I WAS WITH 3
 I SAW 4

QUESTION 2
 SUSPECT (1-5)? 4
 TIME? 6
 SUSPECT 4 AT TIME 6
 I WAS IN ROOM 9
 I SAW 1

QUESTION 3
 SUSPECT (1-5)? 2
 TIME? 6
 I WAS IN ROOM 6
 I SAW 4

QUESTION 4
 SUSPECT (1-5)? 5
 I WAS IN ROOM 1

QUESTION 5
 SUSPECT (1-5)? 3
 TIME? 7
 I WAS IN ROOM 9
 I WAS WITH 2
 I SAW 4

.
 .
 .

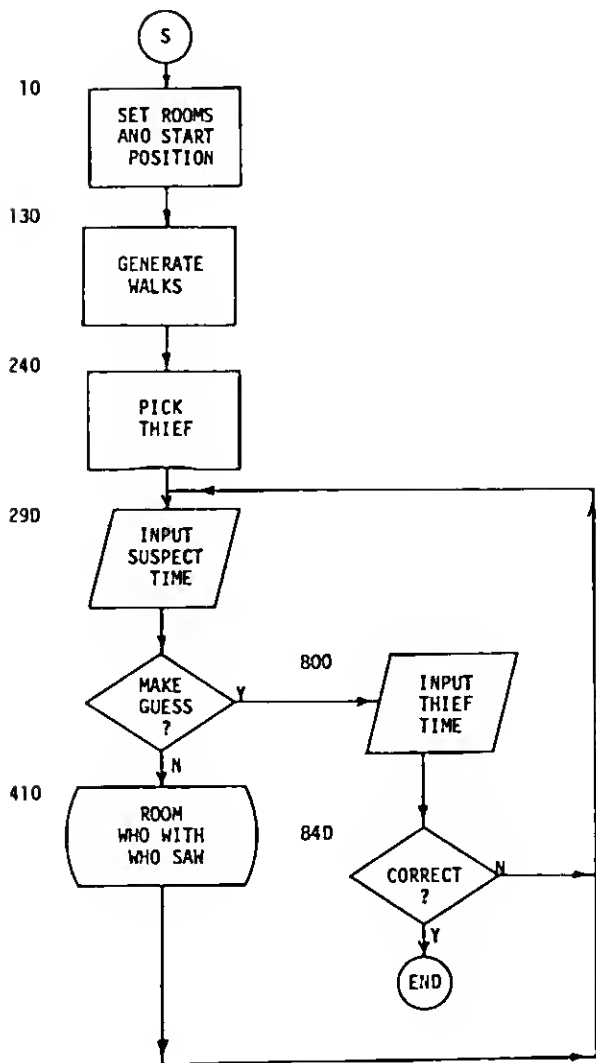
QUESTION 15
 SUSPECT (1-5)? 4
 TIME? 4
 I WAS IN ROOM 5
 I SAW THE DIAMOND
 I WAS WITH 3

QUESTION 16
 SUSPECT (1-5)? 0
 GUILTY SUSPECT? 4
 TIME OF CRIME? 4

YOU GOT 'EM
 THE THIEF IS 4 AT TIME 4.
 YOUR RATING IS 84
 PLAY AGAIN?



DIAMOND THIEF FLOWCHART



DIAMOND THIEF

Variables

A(I,J)	Adjacent rooms
L(I,J)	Room where person I is located at J time
T	Time of theft
O	Thief
P	Probability
S	Suspect
G	Time of guess
A	Temporary variable
I,J,K	Indices

Listing

```

10  DIM A(9,3), L(5,12): Q=1: PRINT "WAIT"
20  FOR I=1 TO 9
30  FOR J=1 TO 3
40  READ A
50  A(I,J)=A
60  NEXT J,I
70  DATA 2,4,0,1,3,0,2,6,0
80  DATA 1,5,7,4,6,8,3,5,9
90  DATA 4,8,0,5,7,9,6,8,0
100 FOR I=1 TO 5
110 L(I,1)=INT(RND(1)*9+1)
120 NEXT I

130 FOR I=2 TO 12
140 FOR J=1 TO 5
150 K=INT(3*RND(1)+1)
160 L(J,I)=A(L(J,I-1),K)
170 IF L(J,I)=0 THEN ISO
180 NEXT J,I

190 T=INT(12*RND(1)+1)
200 FOR I=1 TO 5
210 IF L(I,T)=5 THEN 240
220 NEXT I
230 GO TO 190
240 D=INT(5*RND(1)+1)
250 IF L(O,T)<>5 THEN 240
260 PRINT "SOMEONE STOLE THE DIAMOND."

270 REM START MAIN LOOP
280 PRINT: PRINT "QUESTION": Q
290 INPUT "SUSPECT": S
300 IF S<1 THEN 800
310 IF S>5 THEN 290
320 INPUT "TIME": G
330 IF G<1 OR G>12 THEN 320
340 PRINT: PRINT "SUSPECT": S, "AT TIME": G, ":"
350 IF S=0 THEN P=.5
360 IF S<>0 THEN P=.05
370 IF RND(1)>P OR L(S,G)=5 THEN A=L(S,G): GO TO 410
380 I=INT(3*RND(1)+1)

```

```

390 A=A(L(S,G),I)
400 IF A=0 OR A=5 THEN 380
410 PRINT: PRINT "I WAS IN ROOM"; A
420 IF A<>5 THEN 450
430 IF T<G THEN PRINT " I OIO NOT SEE THE DIAMONO!": GO TO 450
440 PRINT "I SAW THE DIAMONO."
450 IF RND(1)<P THEN S10
460 FOR I=1 TO S
470 IF I=S THEN 500
480 IF L(S,G)<>L(I,G) THEN 500
490 PRINT "I WAS WITH"; I
500 NEXT I: GO TO 540
S10 I=INT(7*RND(1)+1): IF I=S THEN S10
S20 IF I<6 THEN PRINT "I WAS WITH"; I
S40 IF RND(1)<P THEN 640
550 FOR I=1 TO 3
560 A=A(L(S,G),I)
S70 IF A=0 THEN 610
580 FOR J=1 TO S
S90 IF L(J,G)=A THEN PRINT "I SAW"; J
600 NEXT J
610 NEXT I
620 GO TO 700
640 J=INT(10*RND(1)+1)
650 IF J<S THEN PRINT "I SAW"; J
700 IF RND(1)>P THEN 770
710 K=INT(10*RND(1)+1)
720 IF K<6 AND K<>J THEN PRINT "I SAW"; K
770 Q=Q+1: GO TO 280
800 INPUT "GUILTY SUSPECT"; S
810 IF S<1 OR S>5 THEN 800
820 INPUT "TIME OF CRIME"; G
830 IF G<1 OR G>12 THEN 820
840 IF S=0 AND G=T THEN PRINT "YOU GOT 'EM!": GO TO 870
850 IF S=0 OR G=T THEN PRINT "PARTLY RIGHT": Q=Q+10: GO TO 280
860 PRINT "BETTER GIVE UP": Q=Q+100
870 PRINT "THE THIEF IS"; O; "AT TIME"; T
900 PRINT "YOUR RATING IS"; 100-Q
910 INPUT "PLAY AGAIN"; Y$
920 IF Y$="Y" THEN RUN
930 ENO

```

DIAMONO THIEF MODIFICATIONS

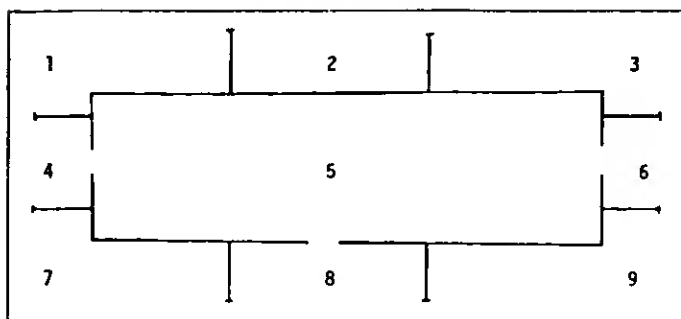
Minor

1. Probability of thief lying -- line 350
2. Probability of innocent suspect lying -- line 360

Major

1. Change room design.
2. Have an accomplice.
3. Jewel is hidden after it is stolen.
4. A guard is roaming around the museum as well.
5. Give suspects and rooms actual names, for example, Mr. Smith is in the Red Room.

MUSEUM FLOOR PLAN

[illegible]